

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

LINCOLN LABORATORY MULTI-PROGRAMMING SUPERVISOR

FRANK BELVIN
JOEL M. WINETT

Group 28

LINCOLN MANUAL 78

6 JANUARY 1967

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-5167.

This report may be reproduced to satisfy needs of U.S. Government agencies.

Distribution of this document is unlimited.

LEXINGTON

MASSACHUSETTS

AD650190

Lincoln Laboratory Multi-Programming Supervisor

Abstract

A supervisor program has been written for the IBM System/360 which allows the multi-programming of several jobs, each with its own input-output requirements.

A job program operates in problem state, and realizes all of its input/output demands through supervisor calls. I/O operations of one job are overlapped with those of other jobs, and can also be overlapped within a single job. Each job is written completely independent of all other jobs. A job is initiated by a request typed on the console typewriter and runs, with frequent I/O waits, until it calls for its own termination, or is terminated by the operator via the console typewriter. During the operation of the supervisor, a job can be initiated, can terminate, and can be re-initiated as often as desired.

All jobs reside in core together with the supervisor. To minimize space, buffers are assigned at the initiation of a job and are released upon termination. Job programs may be written as re-entrant, whereby a single copy of a job program can be active for more than one task. Re-entrant jobs, thus, have different buffer areas for each distinct initiation.

When a job is started, the operator specifies the I/O devices required (e. g. , a particular printer and tape unit for a print job), and any initial parameters desired (e. g. , tape density setting, instructions to skip to a specified listing, instructions to stop after a tape mark, and so forth). The job is then started if the requisite I/O devices and buffers are available. The I/O devices remain assigned to that job until it terminates. However, at any time a job can issue messages to the operator via the console typewriter.

Multi-programming is achieved by central handling of I/O, and by the fact that the jobs run are, in every case, I/O bound. Thus, while a job is waiting on I/O, other jobs can make use of the central processor. When an I/O device has completed its operation, the job which started the I/O operation, through a supervisor call, is permitted to continue.

No standard device error-recovery routines are built into the supervisor, but a family of subroutines is developing which jobs can call upon to provide error recovery. Unit check and unit exception indications are provided to the job at the time the job executes the supervisor call to wait on a device.

The conventions which a job program must follow deal with supervisor call calling sequences since it cannot use privileged instructions. A job controls its input/output by issuing a supervisor call to execute a channel command. (Command chaining, data chaining, and TICing are allowed, but PCI's are prohibited.) If certain suggestions are followed in the writing of channel programs, the system's efficiency is improved; however, the supervisor does no policing of channel programs. Errors committed in the running of a job cause the job to be terminated with a job dump. Storage protection is not used and thus one cannot guard against all errors; hence, all job programs should be debugged before they are included in the system.

Some of the jobs included in the system are:

1. System Jobs - To list the jobs in the system, to list the active jobs, i. e., tasks, or to terminate an active job.
2. Tape Move Jobs - Skips files or records either forward or backward as desired.
3. Prestore Job - Places BCD or binary card images onto tape with standard look-ahead characters.
4. Print Job - Reads, deblocks, and prints a BCD tape with Fortran carriage control characters.
5. Punch Job - Reads, deblocks, and punches card images from tape.
6. Tape Copy and Compare - Copies and compares a mixed mode tape of any density.
7. Tape Dump Jobs - Dumps a 7-track tape in BCD or octal format or a 9-track tape in hexadecimal or EBCDIC.
8. System Dump Jobs - Produces a core dump of a job program, I/O interrupts, or all of core storage as desired.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

Lincoln Laboratory Multi-Programming Supervisor

Frank Belvin

J. Winett

1 April 1967

	<u>No. of Pages</u>
Abstract	2
Acknowledgment	1
I. Supervisor Usage	1
A. Suggested Procedures	1
B. The Distribution Tape	3
C. Loading Procedure	4
D. Functional Description of the Supervisor and Its Subprograms	2
E. Preparing a Supervisor System	1
II. Supervisor Operations	1
A. Loading the Supervisor from Cards	1
B. Loading the Supervisor from a System Tape	1
C. System Control Panel	1
D. Supervisor Wait State	1
E. The Task	1
F. The Job Request	1
G. Initiating a Task	1
H. Correcting a Typing Mistake	1
I. Task Termination	1
J. Sample of System and Task Operations	3
III. Job Operating Instructions	1
A. Job Summaries	1
1. System Jobs	1
2. Dump Jobs	1
3. Utility Jobs	4
4. Job Request Summary	2
5. Sense Switch Usage Summary	1
6. Buffer Usage Summary	2
B. Job Write-Ups	1
1. JOBS, TASKS, TERM, SWITCHES	2
2. JOBDUMP, LOWCORE, COREDUMP	1
3. FSR, BSR, FSF, BSF, REW, RUN, WTM, ERG	4

No. of Pages

III. Job Operating Instructions

B. Job Write-Ups (Cont.)

4. PRESTORE	3
5. PRINT/PUNCH/PUNCHSEP/EJECT	14
6. TPCHECK	7
7. TPCOPY	9
8. T7DUMP	3
9. T9DUMP	2
10. LABEL	2
11. CALCOMP	3
12. CHANGBUF	2
13. DISKDUMP	3
14. HNPLOAD/UCSLOAD	6
15. CONVERT/REPRO	5
16. PUNCHTAP	3
17. LESTER	8
18. LOAD	19
19. UNLOAD	6
20. SHARE	8

IV. The Job Program

A. Writing a Job Program	1
1. The Job Header	2
2. Entry to a Job Program	1
3. The Job List	2
4. Use of Job Sense Switches	1
5. Buffer Sizes	1
B. Supervisor Call Summary	5
C. Supervisor Calls	1
0. JOBDMP	2
1. DMPLST	2
2. STIO	2
3. WAIT	2
4. DORMNT	1
5. RESCNT	1
6. END	1
7. WRITE	1
8. SAVE	1
9. RESTOR	1

	<u>No. of Pages</u>
IV. The Job Program	
C. Supervisor Calls (Cont.)	
10. DELETE	1
11. READ	1
12. SKIP	1
16. SPWAIT	1
18. SAVSKP	1
19. SNSADR	2
20. GETID	1
D. Job Program Subroutines (Unit Check Routines)	1
1. Tape Utilities	
ERG, WTM, FSR, BSR,	
FSF, BSF, REW, RUN	2
2. Card Reader/Punch Unit Check -	
Unit Exception	
CDRDUC, CDSTUC	14
3. Tape Unit Check	
TPWRUC, TPRDUC	9
4. Printer Unit Exception and Unit Check	
PRTUE, PRTUC	10
5. Find Job List	
FJBLST	2

Lincoln Laboratory Multi-Programming Supervisor

Acknowledgment

The authors wish to acknowledge the support given by J. A. Arnow during the design and implementation of this system. It is with his encouragement that we were able to carry out this work which, as well as producing a productive system, gave us the opportunity to learn about the System/360 and to develop ideas on multi-programming.

Job programs to be run under the Lincoln Laboratory Multi-Programming System have been written by:

A. W. Armenti, J. A. Arnow, F. Belvin, N. Daggett,
D. R. Dermenjian, C. E. Durocher, D. E. Hall,
P. Heffernan, H. F. Feinleib, J. F. Nolan, J. R. Walton,
J. M. Winett, and P. M. Wortman.

Lincoln Laboratory Multi-Programming Supervisor

12 April 1966

I. Supervisor Usage

- A. Suggested Procedures
- B. The Distribution Tape
- C. Loading Procedures
- D. Functional Description of the Supervisor and its Subprograms
- E. Preparing a Supervisor System

A. Suggested Procedures

The distribution tape has been designed for use with a System/360 model 40; however, any machine capable of reading a 7-track tape of standard density can be used to obtain printouts of the listings and source decks. All files but the first are recorded with even parity, in standard BCD mode, and at the density specified on the tape label. The first file, however, is recorded at 800 bpi, with data converter on, and contains the object decks and control cards forming a self-loading module. It is intended for use as the operating system.

If a System/360 model 40 is available with devices attached having the same addresses as those in the distribution tape, the system may be loaded by mounting the distribution tape on a tape drive, dialing the load switches to the tape unit, and pressing the IPL button. On normal completion of loading, a message identifying the current system will be printed on the console typewriter. A request can then be made to initiate any desired task. A console listing of all jobs available in the system can be obtained by initiating the 'JOBS' task. A listing of the supervisor and its subprograms, as well as the job programs, can then be obtained by initiating the 'PRINT' task as follows:

```
PRINT PTRp Txxx
```

where 'PTRp' is 'PTRE' or 'PTRF' according to which printer is to be used, and 'Txxx' identifies the tape unit in which the system tape is mounted. Normal completion of printing will be indicated by the message 'END OF JOBS'. At this point, the 'PRINT' task can be terminated using the TERM task as follows, where n is the PRINT task number:

```
TERM n .
```

The source card can then be printed or punched by issuing one of the following tasks:

```
For printing - PRINT PTRp Txxx END
```

```
For punching - PUNCH Txxx END .
```

Again, normal completion of the task will be indicated by the message 'END OF JOBS'.

If the device addresses do not correspond to those of the distribution tape, core patches can be made to alter the device addresses.

If a System/360 model 40 is not available, all files of the distribution tape, except the first should be printed or punched using some other machine.

B. The Distribution Tape

The distribution tape contains 77 files as indicated below. All files, except the first, are recorded with even parity, in standard BCD mode, and at the density specified on the tape label. The first file, however, is recorded at 800 bpi with data converter on. Files 49 and 50 are separated by three tape marks (EOF's).

File 1	Loadable system including BPS loader, BPS single phase dump, supervisor object decks, job object decks, and the necessary BPS control cards for loading into a 65K byte IBM System/360 Model 40.
File 2	Assembly listing of the main supervisor system program, SUPER.
File 3-10	Assembly listing of the 8 supervisor subprograms BUFFER FNDJTL EXIT TABLES FIDCQ JBRP BUFFS FBJTJL
File 11-13	Assembly listing of the job program subroutines TPSUBR CARDUC TAPEUC
File 14	Assembly listing of the TERM Job.
File 15-27	Assembly listing of 13 job programs. JOBS TPMOVE (8) TPCOPY TASKS CARDTP T7DUMP JOBDMP PRINT T9DUMP LOWCOR PUNCH LABEL CORDMP
File 28-49	Assembly listing of the Joblist subprograms for all 20 jobs plus the subprograms FSTJOB and LSTJOB.
File 50	Source cards from the Model 40 Equalities deck.
File 51	Source cards for the main supervisor system, SUPER.
File 52-59	Source cards of the 8 supervisor subprograms.
File 60-62	Source cards of the job program subroutines.
File 63	Source cards for the TERM Job.
File 64-76	Source cards for the 13 job programs.
File 77	Source cards for the Joblist decks including FSTJOB and LSTJOB.

The system, as distributed, is assembled for the Lincoln Laboratory configuration and device addresses. The devices, their addresses, and halfword device address patch locations are listed below.

Device	Address	Relative Patch Address in TABLES Subprogram
Channel 0 Devices		
2540 Card Reader	000C	000880
2540 Card Punch	000D	0008A4
1403 Printer	000E	0008C8
1403 Printer	000F	0008EC
Calcomp Plotter	0020	000910
Calcomp Plotter	0021	000934
2701 Parallel Data Adapter	0022	000958
2741 Typewriter Terminal	FFFF	00097C
2741 Typewriter Terminal	FFFF	0009A0
2741 Typewriter Terminal	FFFF	0009C4
2741 Typewriter Terminal	FFFF	0009E8
2741 Typewriter Terminal	FFFF	000A0C
2741 Typewriter Terminal	FFFF	000A30
2250 Display Scope	FFFF	000A54
2260 Display Terminal	FFFF	000A78
2260 Display Terminal	FFFF	000A9C
Channel 1 Devices		
7-Track Tape	0180	000AC0
7-Track Tape	0181	000AE4
7-Track Tape	0182	000B08
9-Track Tape	0183	000B2C
9-Track Tape	0184	000B50
9-Track Tape	0185	000B74
Channel 2 Devices		
9-Track Tape	0280	000B98
7-Track Tape	0281	000BBC
7-Track Tape	0282	000BE0
7-Track Tape	0283	000C04
7-Track Tape	0284	000C28
7-Track Tape	0285	000C4C

Other fields in the Device Table can be modified by referring to the listing of the TABLES program.

The 2741 Typewriter Terminals, 2260 Display Terminals, and the 2250 Display Scope have not yet been attached to our System/360 Model 40 but have been provided for in the TABLES subprogram. The dummy address of FFFF has been used until we determine what the actual addresses will be.

The console typewriter is not handled like other devices since it is controlled by the supervisor and is used by all job programs. Other devices are not shared and, consequently, are used by only one job program at a time. The address of the console typewriter (TYPAD) is 0009 and is contained in the halfword at relative location 003478 in TABLES.

The supervisor checks that all storage addresses passed to the supervisor through a calling sequence from job programs are within the bounds of the machine core storage. This check is made using the maximum address mask (MAXADD) which is the ones complement of the maximum address. For a machine with 65,536 bytes of core, this mask is X'FFFF0000' and is contained in the word at relative location 003470 in TABLES.

To change the device addresses, the TABLES subprogram can be reassembled, replace cards can be included in the TABLE object deck, or patches can be made in core after the system has been loaded. The relative locations to be modified with replace cards are indicated above. In making a core patch, the location where the TABLE subprogram has been loaded is obtained from the loading storage map and this location is added to the relative locations in TABLES.

The DUMP program has been set to have the dump printed on printer E. If it is desired to modify the printer address on which a dump will be produced, the halfword at relative location 000C62 in the dump program should be changed (either with a replace card or by a patch) to contain the desired printer address.

To make a core patch, the instruction address switches should be set to the location ISTART (obtained from the loading map) and the address compare switch set to stop. Loading is then initiated. During the loading process the machine will stop at the location ISTART. The start button should then be pressed to continue loading. When the machine stops a second time, core patches can then be made, the address compare switch set to process, and the start button pressed to begin operation of the supervisor.

C. Loading Procedure

The LLMPS object decks and loader control cards can be loaded with the IBM Basic Programming Support (8K Card) Utility Program relocating loader (360P-UT-020). Loading can be accomplished from either cards or tape. If it is desired to load the system from tape, the Prestore job can be used with parameter 'EBCD', to place the LLMPS system on tape, thus forming a self-loading 7- or 9-track tape version of LLMPS. Alternatively, the system can be loaded directly from cards. The structure of the system deck is shown in Fig. I. C-1. The storage map obtained on loading the system provided on the distribution tape is shown in Fig. I. C-2. The format of all loader control cards and object deck cards is described in IBM System/360 Basic Programming Support Basic Utilities (C28-6505).

Job programs are placed after the TERM job and may be in any order. The subprograms forming the JOBLST are placed after all job subprograms. The JOBLST begins with the FSTJOB subprogram and ends with the LSTJOB subprogram.

Additional job programs can be inserted among the existing job programs, and require an associated job list entry program (as specified in IV. A-3) to be placed among the existing job list entry subprograms. The size of the job list can be increased by modifying the "ICS JOBLST" card. Each entry in the job list requires 16 bytes and must have a unique subprogram name. The JOBLST for the TERM job has the name FSTJOB. The last job list entry subprogram has the name LSTJOB. Job list entry subprograms for each job should have as subprogram name (symbol field in the start card) a name formed from the author's name and a sequence number, e.g., AUTHR1, AUTHR2, etc.

The setting of the set location counter card preceding the BUFFER program is chosen to place the buffers at the highest location in core. The listing from the BUFFS program is used to determine the total size of the buffers, and this size is then subtracted from the number of bytes of core storage to determine the value in the SLC card. This value must not be less than the address of "FINIS" in program SUPER.

The last card (after super) is an LDT ISTART card and is placed to begin supervisor operation.

The lowest location used for loading is 4096 (hex 1000) to provide room for storing I/O OLD PSW, CSW, and other information useful during debugging. This location can be lowered by modifying the constants LOWINT and HIINT defined in the equalities and used in the main "SUPER" program.

				ID Field
IBM BPS Low Relocating Loader - Version 2 (360-UT-020). See Note 1.				40 LO
SLC	00C500		Set to position buffers above SUPER in high core	
	BUFFER		Subprogram to specify buffer areas	BUFF
SLC	001000		Start loading at X'1000'	
IBM BPS Single Phase Dump - Version 1 (360-UT-019A). See Note 2.				DMPS
ICS	DMPEND	0008	Remembers Location Counter	
SLC	000C6C	DUMP	See Note 3	
ICS		SNTPIN		
SLC	000BE0	DUMP		
ICS		OUTDEV		
SLC		DMPEND	Reset Location Counter	
ICS		JOBLST	01A0 Save space for 26 Job List Entries	
	TABLES		Supervisor system subprograms	40TAB
	BUFFS		Supervisor system subprograms	40BUF
	FNDJTL		Supervisor system subprograms	FNDJT
	FIDCQ		Supervisor system subprograms	FIDCQ
	FBJTJL		Supervisor system subprograms	FBJTJL
	EXIT		Problem state subprogram	EXIT
	JBRP		Problem state subprogram	JBRP
	TPSUBR		Job program subroutine	TMSR
	CARDUC		Job program subroutine	CARDU
	TAPEUC		Job program subroutine	TAPEU
	TERM		Job program	TERM
	JOBS		Job program	JOBS
	TASKS		Job program	TASKS
	JOBDMP		Job program	JBDMP
	LOWCOR		Job program	LOW
	CORDMP		Job program	CORE
	TPMOVE		Job program	TMJB
	CRDTAP		Job program	CRDTP
	PRINT		Job program	PRINT
	PUNCH		Job program	PUNCH
	TPCOPY		Job program	TPCPY
	T7DUMP		Job program	T7DMP
	T9DUMP		Job program	T9DMP
	LABEL		Job Program	TPLBL
ICS	SSUPER	0008	Remember Location Counter	
SLC	JOBLST		Reset Location Counter	
	FSTJOB		Job List Entry Subprogram for TERM job	FSTJB
	Other job list entry subprograms			JBLST
	LSTJOB		Last job list subprogram	LSTJB
SLC	SSUPER		Reset Location Counter	
	SUPER		Main Supervisor State Program	40SUP
LDT	ISTART		Placed to begin operation at 'ISTART' in SUPER	

Fig. I. C-1. LLMPS Load Deck Structure

Note 1:

The loader is the standard IBM BPS High Relocating Loader - Version 2. The End Card Option fields have been left blank which results in loading from the IPL device and gives no storage map or error printout.

Note 2:

Seven Replace cards (REP) have been inserted between the last Relocation Directory Card (RLD) and the End card (END) of the dump program as follows:

REP	000C58	010000, FFF8	Core size
REP	000C5C	01FFFF	
REP	000C62	01000 F	Output Printer F
REP	000C62	01000 E	Output Printer E
REP	000F30	010200, 0000, 0000	Steal I/O Interrupt
REP	0010FA	01D200, 0000, 0000	Steal I/O Interrupt
REP	000E62	014580, F1B0	Kill typewriter

The last three Replace cards, as well as the Set Location Counter cards (SLC) and Include Control Section cards (ICS) that immediately follow the dump program, have been added to enable the LLMPs supervisor to handle all I/O interrupts even when the dump program is being executed.

The dump will be printed on Printer E. If it is desired to have the dump on Printer F, the third and fourth should be interchanged. Alternatively, a different REP card can be used for a different printer address.

Note 3:

The ICS and SLC cards following the dump program are included to define the entry points SNTPIN and OUTDEV in the dump. The symbol DMPEND defines the double word aligned location after the dump subprogram and is used to reset the location counter after the dump entry points have been defined.

STORAGE MAP

BUFFER	AT LOCATION	000500
DUMP	AT LOCATION	001000
TABLES	AT LOCATION	002260
JOBTBL	AT LOCATION	002260
DEVLST	AT LOCATION	002990
DEVTBL	AT LOCATION	002AFC
PCCL	AT LOCATION	002EDC
MAXACC	AT LOCATION	0036DC
NUMDEV	AT LOCATION	0036D4
TYPAD	AT LOCATION	0036D8
BUFFS	AT LOCATION	0036LC
BUFMAX	AT LOCATION	0036E0
BUFSIZ	AT LOCATION	0036F8
BUFLST	AT LOCATION	00371C
ENCBLS	AT LOCATION	003764
FNDJTL	AT LOCATION	003768
FIDCC	AT LOCATION	0037B8
FBJTJL	AT LOCATION	00381C
EXIT	AT LOCATION	0039E8
JBRP	AT LOCATION	003A0C
TPSUER	AT LOCATION	003DEC
WTM	AT LOCATION	003DE8
ERG	AT LOCATION	003E64
FSR	AT LOCATION	003EA4
BSR	AT LOCATION	003EF8
BSF	AT LOCATION	003F8C
FSF	AT LOCATION	003F38
REW	AT LOCATION	003FC4
RUN	AT LOCATION	0040C4
CARDUC	AT LOCATION	004160
CDSTUC	AT LOCATION	004160
CCRDUC	AT LOCATION	004176
TAPEUC	AT LOCATION	004450
TPREUC	AT LOCATION	004450
TPWRUC	AT LOCATION	00479E
TERM	AT LOCATION	004A0C
JOBS	AT LOCATION	004A38
TASKS	AT LOCATION	004B58
JCBDCP	AT LOCATION	004C60
LCWCOR	AT LOCATION	004CF8
CORCMP	AT LOCATION	004D2C
TPMOVE	AT LOCATION	004D6C

JBWIM	AT LOCATION	004D6C
JEERG	AT LOCATION	004DA4
JBBSR	AT LOCATION	004DE8
JBFSF	AT LOCATION	004E28
JBBSF	AT LOCATION	004E64
JBFSR	AT LOCATION	004EAC
JBREW	AT LOCATION	004EEC
JBRUN	AT LOCATION	004F1C
CRDTAP	AT LOCATION	005288
PRINT	AT LOCATION	0057C8
PUNCH	AT LOCATION	0068AC
TPCOPY	AT LOCATION	007838
T7DUMP	AT LOCATION	0089C8
T9DUMP	AT LOCATION	0093E8
TPLAEL	AT LOCATION	009B9C
FSTJOB	AT LOCATION	0020CC
WINET1	AT LOCATION	0020D0
WINET2	AT LOCATION	0020E0
WINET3	AT LOCATION	0020FC
WINET4	AT LOCATION	002100
WINET5	AT LOCATION	00211C
ARM1	AT LOCATION	00212C
ARM2	AT LOCATION	00213C
ARM3	AT LOCATION	002140
ARM4	AT LOCATION	00215C
ARM5	AT LOCATION	002160
ARM6	AT LOCATION	00217C
ARM7	AT LOCATION	00218C
ARM8	AT LOCATION	00219C
FRANK1	AT LOCATION	0021A0
JACK1	AT LOCATION	0021BC
CLARE2	AT LOCATION	0021CC
JACK8	AT LOCATION	0021DC
DIANA	AT LOCATION	0021E0
DD9TRK	AT LOCATION	0021FC
CDLABL	AT LOCATION	00220C
LSTJOB	AT LOCATION	00221C
SUPER	AT LOCATION	009EB8
ISTART	AT LOCATION	00B218
KILCHR	AT LOCATION	00A339
BRKCHR	AT LOCATION	00A331
DELCHR	AT LOCATION	00A747
JOBNC	AT LOCATION	00A5B4
FINIS	AT LOCATION	00B843

Fig. I. C-2. Storage Map for System/360 Model 40
LLMPS with Jobs as of 1 April 1966.

D. Functional Description of the Supervisor and its Subprograms

The LLMPS program consists of thirteen programs which are named in accordance with the symbol appearing in the name field of the START card of the corresponding source deck. These programs are listed below, with a brief description of their function.

1. SUPER. This is the basic supervisor program. It operates in the Supervisor state, and handles all interruptions, manages all I/O operations and CPU resource allocation, and provides special services to the tasks operating under LLMPS.

2. TABLES. This program contains no executable code. It serves to reserve space for and set values in several tables used by the supervisor to manage its operations. The tables have been separated from the SUPER program to facilitate making modifications. These tables include the job tables (JOBTBL), the device list (DEVLST), and the device tables (DEVTBL).

3. BUFFS. This program contains no executable code. It is a directory of buffer (free storage) areas. It is used during task initiation to assign buffers to a task, and at task termination to return the buffers to free storage.

4. BUFFER. This program contains no executable code and reserves no storage. It is used to indicate the beginning of the buffer area, the directory for which is maintained in BUFFS.

5. DUMP. This is the IBM-supplied single-phase dump program, modified as shown in "Loading Procedures". It operates in supervisor state, and is the only program besides SUPER which does so. It is used to provide diagnostic dumps of the supervisor or its tasks.

6. FNDJTL. This problem state subprogram converts the task number to the corresponding job table location.

7. FIDCQ. This problem state subroutine frees the I/O devices and clears the CPU queue for the specified task. It is used only to terminate a task.

8. FBJTJL. This problem state subroutine returns buffers to free storage and clears the job list entry (if necessary) at task termination. It also prints the task termination message on the console typewriter.

9. EXIT. This problem state program calls FIDCQ and FBJTJL and terminates a task. It is invoked from a job by executing SVC END.

10. JBRP. This problem state program initiates tasks and formats the parameters typed on the console typewriter.

11. TERM. This problem state program is a job program and is invoked from the console by typing 'TERM n' where 'n' is the task number. It is used by the operator to terminate a task. It calls FNDJTL, FIDCQ and FBJTL.

12. FSTJOB. This program contains no executable code. It is the job list entry for the TERM job and must be the first entry in the job list.

13. LSTJOB. This program contains no executable code. It contains a dummy job list entry and must be the last entry in the job list.

E. Preparing a Supervisor System

The Lincoln Laboratory Multi-Programming Supervisor is formed from the main supervisor program SUPER, the table subprograms TABLES and BUFFS, the supervisor subprograms FNDJTL, FIDCQ, and FBJTJL, and the problem state subprograms EXIT and JBRP. The operation of the supervisor is controlled through the use of a set of tables, the Job List, the Job Table, the Device List, and the Device Table. The format of these tables are defined using equalities and all reference to the fields in a table are made symbolically. The equalities specifying the table formats are defined in an Equalities deck and this deck is included in the assembly of those programs which refer to the tables. Besides defining the table formats, the Equalities deck indicates the number of job tables NJBTBL, the number of devices on each channel NCHNLx, and the total number of devices NDEVS. These numbers must reflect the organization of the tables defined in the TABLES subprogram. The Equalities deck also assigns a symbolic definition to the permanent storage locations in low core and defines the general registers used by the supervisor. In addition, the low and high storage locations used to store the IOO PSW and CAW on an I/O interruption are defined in the Equalities deck.

The TABLES subprogram defines the tables used by the supervisor. It contains the Job Tables, the Device List, and the Device Tables, as well as a mask defining the core size, the device address of the console typewriter, the total number of devices and a Pool of 256 double word entries used for the device I/O queues and the CPU queue. The number of Job Tables defines the number of tasks that can be run concurrently in the system. The Device List and Device Tables indicate the devices available to the system and must correspond and be grouped by channels.

The BUFFS subprogram defines the set of buffers available for allocation to a job program and also indicates the size of the largest buffer. The number of buffers of a particular size and the sizes are specified as equalities. The BUFSIZ table contains double word entries for each buffer size. The BUFLST table contains a four byte entry for each buffer to be defined. The BUFLST table must immediately follow the BUFSIZ table. The storage area reserved for the buffers is specified by the location of the BUFFER subprogram which should be located in high core above the SUPER program.

To assemble the source decks, the Equalities deck must be included in the source decks of the SUPER, TABLES, and JBRP supervisor programs, and the source decks of the JOBS, TASKS, and JOBDMP job programs. The distribution tape has the Equalities for the Lincoln Laboratory configuration inserted in these decks. The assembly procedure can be performed from the source program decks which are stored on the distribution tape. The object decks produced can be used to form a system by placing the SUPER object deck (the first object deck produced) after the other object decks, placing a loader and the dump subprogram at the front of the deck, and inserting the appropriate control cards.

Lincoln Laboratory Multi-Programming Supervisor

11 April 1966

II. Supervisor Operations

- A. Loading the Supervisor System from Cards
- B. Loading the Supervisor from a System Tape
- C. System Control Panel
- D. Supervisor Wait State
- E. The Task
- F. The Job Request
- G. Initiating a Task
- H. Correcting a Typing Mistake
- I. Task Termination
- J. Sample of System and Task Operations

A. Loading the Supervisor from Cards

To load the supervisor system from cards, place the system deck in the card reader. Ready the card reader and press the End of File (EOF) button. Set the load switches to the address of the card reader and then press the LOAD button.

All cards should be read, the machine automatically placed in Wait State, and the name and date of the current system printed on the console typewriter. The system is then ready for operation. If the loader encounters any trouble, an operator message code may be printed on the console typewriter and displayed in the instruction address lights. The codes and their meaning are described in Appendix A, Form C28-6557, BPS Operating Guide for Basic Assembler and Utilities.

B. Loading the Supervisor from a System Tape

To load the supervisor from the system tape, mount the system tape on a tape drive and ready the unit. Dial the load switches to the address of the tape unit. Press System Reset and then press the Load button. The supervisor will be read from the system tape, loaded into core, and ready for use.

C. System Control Panel

During the normal operation of the supervisor, the Interrupt button on the System Control Panel has no effect. It may be pressed at will. When there is a demand for the console typewriter but it is in a not ready state, the machine is put in Wait state with code F0F0F0 in the current instruction address. When this happens, the typewriter should be made ready, and the machine interrupt button pressed to resume operation of the supervisor.

During the operation of the supervisor, the operator control panel on the Model 40 should be set as follows:

- a) Rate switch set to Process.
- b) Address Compare switch set to Process.
- c) CPU check switch set to Stop.
- d) Interval timer disabled.

D. Supervisor Wait State

When the supervisor is in operation, it may be in either Wait or Run state. When it is in Wait state, the code in the current instruction address indicates the reason for being in Wait state. Under normal conditions, the supervisor goes into Wait state whenever there is no demand for the central processor, although I/O operations may be in progress. In this case, the code in the instruction address is 000FF0.

The supervisor may go into Wait state under three other conditions. When there is a demand for the console typewriter but it is not in a ready state, the system alarm will sound and the machine put in Wait state with the code F0F0F0 in the instruction address.

The typewriter should be made ready and the machine interrupt button pressed. If the interrupt button is pressed while the typewriter is not ready, the alarm will be sounded again.

When the console typewriter is in CE mode, the alarm cannot be sounded even though the typewriter is not ready for use. This condition can be recognized by the Wait state code F0F0F0 in the instruction address.

Another condition in which the supervisor will go into Wait state is when there is an unrecoverable supervisor error. The code under this condition is FFFFFFFF. If this should occur, a core dump should be taken, and the system reloaded. This dump should be saved and the responsible system programmer notified.

If a machine check should occur, the Supervisor will go into Wait state with code 000AAA. The IBM CE should be called immediately.

E. The Task

A task is created on the activation of a job program by a request issued through the use of the console typewriter. Associated with each task is a task number, assigned at the creation of the task, used to identify the task. Through the use of the task number, the pseudo sense switches associated with a task can be read and altered (SS job), the task can be terminated from the console typewriter (TERM job), or a dump of the job program together with the tables associated with the particular task can be produced (JOB_DUMP job). The task number is incremented as each new task is created.

F. The Job Request

A request consists of a job name, device ID's, if required, and a variable number of parameters (possibly none) separated by any number of spaces. For each job, the required ID's and the optional parameter are described in a write-up for the job. A request may be typed using either upper or lower case alphabetic character.

The device ID's, if required, must be specified in a particular order. The precedence of devices are indicated in the list below. For example, a printer ID must be specified before a tape ID.

<u>Device Name</u>	<u>ID</u>	<u>Type</u>
RDR	RDRr	2540 Card Reader
PCH	PCHc	2540 Card Punch
PTR	PTRp	1403 printer
7TP	Txxx	7-track tape unit
9TP	Txxx	9-track tape unit
TP	Txxx	either 7-track or 9-track tape unit
DISK	Dyyy	2311 disk unit
PDA	PDA _n	2701 Parallel Adapter
PLOT	PLT _n	Calcomp Plotter
SCOP	SC _n	2250 Display Scope
DISP	DT _n	2260 Display terminals
TYPE	Xxxx	2741 Typewriter terminal
TTY	TTY _n	Teletype terminal
CONS	CONS	1050 Console typewriter

If only one unit of a given device type is attached to the system, then the device identification of the unit is not specified in the job request.

The parameters for a job may be specified in any order except that parameter pairs must be specified together. A parameter may be either alphanumeric or numeric. If a parameter is alphanumeric (i.e., contains a character other than a digit or a plus or minus sign) only the last 4 characters of the parameter field are considered as the parameter. If a parameter is numeric, it may be a signed integer of up to 8 digits. If the numeric parameter contains more than 8 digits, only the last 8 digits are considered as part of the parameter. If a numeric field contains an embedded plus or minus sign only those digits following the sign are considered as forming the numeric parameter.

G. Initiating a Task

To initiate a task, i.e., to activate a job, the Request Key on the console typewriter should be pressed. A task number to be associated with this task will be printed. A job request should then be typed and the request line terminated with a carrier return or End of Block (ALT together with EOB) code.

To cancel a request and to free the typewriter, type the cancel code (ALT together with CANCEL). To initiate another request, the Request Key must be pressed again.

If more than the maximum number of character permitted per line are typed (120), or more than the maximum number of parameters for a job are specified (13), the task will not be initiated and a message will be printed.

Under certain timing situations an overrun condition can occur for the console typewriter. In this case, a message is printed, and the task number is printed again so that the task can be reinitiated.

H. Correcting a Typing Mistake

A typing mistake can be corrected either on a character basis or on a line basis or, in some circumstances, on a parameter basis.

a. On a character basis:

Type a quote character (") for each single character or space to be deleted. To delete the last n characters or spaces, type n quote characters (e.g., to delete the last 3 characters, type 3 quote characters).

b. On a line basis:

Type a question mark (?) to delete a complete line and begin again on the next line. The task number is automatically retyped.

c. On a parameter basis:

If a mistake is made in typing a four character alphanumeric parameter, more than four characters may be typed in this field where the last four characters form the intended parameter. No blank (or space) characters may be part of a field.

If a mistake is made in typing a numeric parameter, the intended numeric parameter may be typed as the last 8 digits of the field (possibly with leading zeros) if the sign of the integer thus formed is correct. Alternatively, the sign of the parameter may be retyped followed by the digits of the number, since only those digits following the sign are considered as forming the numeric parameter. No blank (or space) characters may be part of a field.

I. Task Termination

Whenever a task terminates, a message is printed on the console typewriter giving a termination code for the task. The termination codes are:

Associated with Task --

- | | |
|-------|---|
| O. K. | Satisfactory completion of the task. |
| KILL | Task terminated by console initiation. |
| ERR | Request error because of one of the following conditions:

JOB NAME NOT FOUND
JOB NOT FREE
DEVICE NAME NOT FOUND
DEVICE NOT FREE
DEVICE NOT OPERATIONAL
NO BUFFER AVAILABLE
BUFFER SIZE TOO LARGE
JOB LIST ERROR
JOB HEADER ERROR |
| INOP | Device inoperational. Check for control unit off-line. |

Associated with Job Program --

- | | |
|------|---|
| PGNT | Program error interrupt in job program. |
| SVC | Supervisor call error. |
| ILDN | Illegal logical device number in job program. |
| CSW | Channel program error in job program. |

If the TERM job is used to terminate a task, two messages will be printed; one for the task being terminated (code: KILL) and one for the TERM task (code: O. K.).

The termination codes associated with a job program should not occur except while a job program is being debugged. When they do occur, a dump of the job program will have been produced. The dump, together with the console listing, should be saved for the job programmer.

J. Sample of System and Task Operations

The attached printout from the console typewriter illustrates the operation of the Lincoln Laboratory Multi-Programming Supervisor. On the first initiation of task 1, the question mark (?) was used to cancel the line typed and to begin the task again. The PRINT job was then initiated as task 1. On the initiation of task 2, the quotation character (") was used to delete the erroneously typed character (r). This job, JOBS, list all the jobs in the system together with the required devices. A tape copy job was then initiated as task 3 to copy tape T183 onto tape T184. The TASKS job, task 4, list the active jobs and the device units being used. Notice the message written by the task 3 TPCOPY job after the information from the TASKS job.

The PRESTORE job was initiated as task 5 using tape unit T281. Since there is only one 2540 card reader attached to the system, the identification of this device was not specified in the job request. Tape T280 was rewound as task 6 and tape T283 was rewound as task 7. Note here the use of the quotation mark (") to change the specified tape from 286 to 283. In task 8 the TPCOPY was initiated with optional parameters PAR and EVEN. The messages printed after this initiation come from tasks 1, 3 and 8 as indicated.

As task 9, an undefined job (CANCEL) was initiated. This task was terminated with code ERR. On task 10 an overrun condition occurred.

Fig. II.J-1

MULTI-PROGRAMMING SUPERVISOR VERSION 04/01/66

```

00001  jobs?

00001  print ptre t281

00001      PRINT    TAPE TO PRINT (JA) VERSION 5
00001      PRINT    END OF FILE

00002  jobsr"

00002      JOBS      (R)  INDICATES RE-ENTRANT JOB
00002      JOBS      VERSION JOBNAME      REQUIRED DEVICES
00002      JOBS      (P)  a      TERM
00002      JOBS      (R)  1  COREDUMP
00002      JOBS      (R)  1  JOBDUMP
00002      JOBS      1      JOBS
00002      JOBS      1      TASKS
00002      JOBS      1      LOWCORE
00002      JOBS      (P)  a      PSR      TP
00002      JOBS      (P)  a      PSF      TP
00002      JOBS      (R)  a      ESF      TP
00002      JOBS      (P)  a      ESR      TP
00002      JOBS      (P)  a      REW      TP
00002      JOBS      (P)  a      RUN      TP
00002      JOBS      (P)  a      WTM      TP
00002      JOBS      (P)  a      ERG      TP
00002      JOBS      3  PREFSTORE  PDR      TP
00002      JOBS      (P)  5  PRINT    PTR      TP
00002      JOBS      1  PUNCH     RDR      PCH      TP
00002      JOBS      (P)  3  TPCOPY   TP      TP
00002      JOBS      1  T7DUMP     PTR      7TP
00002      JOBS      1  T9DUMP     PTR      9TP
00002      JOBS      1  LABEL      RDR      TP
00002      JOBS      JOB 00002 TERMINATED.      CODE 0.K.

00003  tpcopy t183 t184

00003      TPCOPY    OUTPUT IS ON 9-TRACK UNIT
00003      TPCOPY    INPUT  IS FROM 9-TRACK UNIT
00003      TPCOPY    TAPE COPY VERSION 3(1-66) INPUT ---- OUTPUT ----
00003      TPCOPY    COPY PHASE

00004  tasks

00004      TASKS      ACTIVE JOBS      DEVICE UNITS BEING USED
00004      TASKS      00001      PRINT    PTR      T281
00004      TASKS      00003      TPCOPY    T183      T184
00004      TASKS      00004      TASKS
00004      TASKS      JOB 00004 TERMINATED.      CODE 0.K.
00003      TPCOPY    END OF FILE 0001,NO.RCS 00001205

```

```

00005    prestore t281

00001        PRINT    END OF FILE
00005    PRESTORE    DEVICE NOT FREE
00005    PRESTORE    JOB 00005 TERMINATED.        CODE    ERR
00001        PRINT    END OF FILE
00001        PRINT    END OF FILE
00001        PRINT    END OF FILE
00001        PRINT    END OF FILE
00001        PRINT    END OF FILE

00006    rew t280

00001        PRINT    END OF FILE
00001        PRINT    END OF FILE

00007    rew t286"3

00006        REW      JOB 00006 TERMINATED.        CODE    O.K.
00003        TPCOPY   END OF FILE 0002 _NO.RCS=00003151

00008    tpcopy t283 t280 par even

00003        TPCOPY   END OF FILE 0003 _NO.RCS=00000007
00001        PRINT    END OF FILE
00008        TPCOPY   DEVICE NOT FREE
00008        TPCOPY   JOB 00008 TERMINATED.        CODE    ERR
00001        PRINT    END OF FILE
00003        TPCOPY   END OF FILE 0004,NO.RCS=00000640
00003        TPCOPY   END OF FILE 0005,NO.RCS=00000154
00003        TPCOPY   END OF FILE 0006,NO.RCS=00000060
00003        TPCOPY   END OF FILE 0007,NO.RCS=00000063
00003        TPCOPY   END OF FILE 0008,NO.RCS=00000180
00003        TPCOPY   END OF FILE 0009,NO.RCS=00000037
00001        PRINT    END OF FILE

00009    cancel

00007        REW      JOB 00007 TERMINATED.        CODE    O.K.
00009        CANCEL   JOB NAME NOT IN JOB
00009        CANCEL   JOB NAME NOT IN JOB LIST.
00003        TPCOPY   END OF FILE 0010,NO.RCS=00000598
00009        CANCEL   JOB 00009 TERMINATED.        CODE    ERR

00010

00001        PRINT    END OF FILE
00003        TPCOPY   END OF FILE 0011,NO.RCS=00000279
00001        PRINT    END OF FILE

00010    abcdefghijklmnopq
OVERUN, PLEASE TYPE SLOWER.

```

Fig. II. J-1 (Cont.)

Lincoln Laboratory Multi-Programming Supervisor

13 April 1966

III. Job Operating Instructions

A. Job Summaries

1. System Jobs
2. Dump Jobs
3. Utility Jobs
4. Job Request Summary
5. Sense Switch Usage Summary

B. Job Write-Ups

A. Job Summaries

The following brief description of the LLMPs system jobs, dump jobs, and utility jobs indicate the programs available for use, and indicate the date of the latest complete write-up on each program. An asterisk will appear before the jobname if the program or complete job write-up has been changed since the summary sheet was last prepared.

The complete job write-up prepared for each job in the system, contains the following information:

- a) Job Summary
- b) General Information
- c) Task Request Summary
- d) Task Request Format
- e) Operating Notes
- f) Example of the Job Request
- g) Console Messages

The Job Request Summary gives the Jobname, Device ID's, and Parameters for all jobs presently included in the system. Optional parameters are enclosed in brackets with the default value indicated in parentheses. In the cases where the default value cannot be specified as a parameter, the default condition is indicated in lower case. Re-entrant jobs are indicated by (R) preceding the jobname. For each job, the devices used and the possible parameters are described in a separate write-up for the job. The system output unit (indicated by an asterisk on the Job Request Summary) is used for system dump jobs. On the Model 40 the system output unit is Printer 3 (PTR3).

The Sense Switch Usage Summary indicates the usage of the job sense switches. For each job which uses the sense switches, an X appears in the appropriate row indicating that the corresponding sense switch bit is applicable to that job.

The parameter variables used in the job summary are as follows:

```

      tttt  task number
n, n1, n2, ...  count numbers
      xxxxxx  input reel number
      yyyyyy  output reel number
s, nnnnnn  buffer sizes
      jjjj  job number
      jnjnjn  job name
s1, s2, ... s32  bit numbers

```

Lincoln Laboratory Multi-Programming Supervisor

Job Summary

System Jobs

JOBS - 15 July 1966

Prints on the console typewriter a list of the jobs which can be initiated. The required devices for each job are indicated following the jobname; and a version code precedes the jobname. If a job is re-entrant, i.e., can be activated for more than one task, the symbol (R) precedes the version code.

TASKS - 15 July 1966

Prints on the console typewriter a list of the jobs which are active, (i.e., tasks) indicating the device units being used and the task number for each active job.

TERM - 15 July 1966

Terminates an active task. The device units associated with the task are freed and become available for other tasks. This job is re-entrant and may be re-issued to terminate more than one task.

CHANGBUF - 6 June 1966

Performs the task of changing the buffer sizes used by job.

SWITCHES - 15 July 1966

Displays, or modifies and displays, the sense switches of an active task.

Lincoln Laboratory Multi-Programming Supervisor

Job Summary

Dump Jobs

JOB_DUMP - 16 March 1966

A dynamic dump of an active job is produced on the system output unit. This job is re-entrant.

LOWCORE - 16 March 1966

A dump of the first 1000 (hex) bytes of core is produced on the system output unit.

COREDUMP - 16 March 1966

A dump of core is produced on the system output unit. This job is re-entrant.

Lincoln Laboratory Multi-Programming Supervisor

Job Summary

Utility Jobs

1 April 1967

FSR - 16 March 1966

Forward spaces n records of a tape.

BSR - 16 March 1966

Backspaces n records of a tape.

FSF - 16 March 1966

Forward spaces n files of a tape.

BSF - 16 March 1966

Backspaces n files of a tape.

REW - 16 March 1966

Rewinds a tape.

RUN - 16 March 1966

Rewinds and unloads a tape.

WTM - 16 March 1966

Writes n tape marks (end of files) on a tape.

ERG - 16 March 1966

Erases n record gaps (3'' of tape/gap) of a tape.

* CONVERT - 3 November 1966

Accepts cards punched with the codes produced by an IBM Model 026 card punch (the pre-360 BCD code) and produces new cards, on a one-for-one basis, that are punched in accordance with the EBCDIC code used by S/360. This job is useful in converting cards punched on the Model 026 Card Punch to cards that would have been punched on the (newer) Model 029 Card Punch if keys with the same graphics had been used.

*REPRO - 3 November 1966

Produces new cards on a one-for-one basis. The reproduced cards are identical to the input cards regardless of the codes used. Input codes are not checked for validity; therefore, the job may be used to duplicate any deck.

PRESTORE - 27 May 1966

Performs the task of recording card images on magnetic tape. The program runs in two modes; for the 7094 it will prestore both BCD and binary card images adding the proper look-ahead bits to each record, for the 360 it will prestore EBCDIC card images in odd parity with the converter on.

*PRINT - 1 April 1967

Processes labeled and unlabeled 7 or 9 track tapes. 7-track tapes may be read under any density and mode set specification. 9-track tapes force 800 bpi and odd parity. Mixed density tapes cannot be printed.

*PUNCH - 1 April 1967

Punches a 7-track tape written in odd and/or even parity or a 9-track tape. Cards can be punched as column binary, BCD, or EBCDIC. Tapes may be labeled or unlabeled and may be of any density.

*PUNCHSEP - 1 April 1967

Like the PUNCH job except that two blank cards from the reader are inserted into the punched deck whenever an end of file or system output flip card is recorded on the tape.

*EJECT - 1 April 1967

Restores and/or spaces the paper on a printer.

*TPCHECK - 1 April 1967

Examines a 7-track or a 9-track tape for parity errors and converter checks, lists sense information on a permanent read error, switches parity from even to odd (or odd to even) if necessary, recording the location and number of switches, and determines the number of bytes per record and records per file.

*TPCOPY - 1 April 1967

Performs copy and/or comparison of 7- or 9-track tapes. Options are available for selecting parity, density, number of files, and use of the data conversion feature.

T7DUMP - 27 May 1966

Dumps a 7-track tape of either even or odd parity in octal format. This job replaces the function previously performed on the 1401 with the BCD/Binary dump tape program.

T9DUMP - 31 August 1966

Dumps a 9-track tape or a 7-track tape (recorded with density 800, odd parity, converter on) generated on the 360. The dump can be in hexadecimal or hexadecimal and EBCDIC, and the number of files to be dumped may be specified.

LABEL - 5 October 1966

Performs the task of writing a label on a 7-track tape (even parity, translator on, density 200, 556, or 800) or on a 9-track tape (odd parity, density 800) on the 360. The label record is a single record followed by a tape mark (EOF).

CALCOMP - 27 May 1966

Performs the task of plotting standard Calcomp output tapes produced on the 7094 or 360. Each picture or reference point on the tape is separated by a header record. Thus, the operator may skip to any specific picture within a file. Can only be used on the 360/40.

DISKDUMP - 17 June 1966

Displays the contents of a 2311 disk on a printer. Records on the disk may be dumped in both hexadecimal and EBCD or, alternatively, the standard Volume Table of Contents (VTOC) may be displayed. Can only be used on the 360/67.

HNPLOAD - 5 October 1966

Loads the arrangement of 240 codes for either an HN or PN train into the read/write storage units of the 2821 control unit for a 1403 printer.

UCSLOAD - 5 October 1966

Loads a specified arrangement of 240 codes selected from the Extended Binary Coded Decimal Interchange Code into the read/write storage units of the 2821 control unit for a 1403 printer.

* PUNCHTAP - 21 November 1966

Reads card images in EBCDIC code from a 9-track tape (or from a 7-track tape with density 800, odd parity, converter on) and punches them out on paper tape in FIELDATA code. The paper tape produced can be used as input to the Univac 1219.

* LESTER - 14 November 1966

Receives LES telemetry data via one of the LES telephone lines from the tracking station; formats the input data into printable records; and outputs these records onto either a 7 or 9-track tape.

* LOAD - 1 April 1967

Relocates and loads assembled LLMPS programs and program joblists into core.

* UNLOAD - 1 April 1967

Removes from core storage the job programs and joblist programs loaded into core by the LOAD program.

* SHARE - 1 April 1967

List and punch catalog and/or order cards from a SHARE Distribution Tape Reel (DTR) or will retrieve source and object decks from the DTR or from a SHARE Series Library Tape (SLT).

Lincoln Laboratory Multi-Programming Supervisor

Job Request Summary

1 April 1967

	<u>JOBNAME</u>	<u>UNITS</u>	<u>PARAMETERS</u>
	JOBS	-	
	TASKS	-	
(R)	TERM	-	tttt
	SWITCHES	-	tttt [s1 s2 ... s32]
	CHANGBUF	-	jnjnjn [BFNO n] [SIZE nnnnn] [BFNO 1] [SIZE maxbuf]
	LOAD	RDRr	
	UNLOAD	-	
(R)	JOBDUMP	*	tttt
	LOWCORE	*	
(R)	COREDUMP	*	
(R)	ERG	Txxx	[DEN 2] [DEN 5] [n] [DEN 8] (1)
(R)	WTM	Txxx	
(R)	FSR	Txxx	
(R)	BSR	Txxx	
(R)	FSF	Txxx	
(R)	BSF	Txxx	
(R)	REW	Txxx	
(R)	RUN	Txxx	
(R)	CONVERT	RDRr	PCHc
(R)	REPRO	RDRr	PCHp
	PRESTORE	RDRr	Txxx [DEN 2] [DEN 5] [BCD] [MORE] [DEN 8] (EBCD) (terminates)
			[DEN2] [OP] [DEN5] [OT] [94TP] [UFRM] [DEN8] [OC] [NLBL] [VFRM] [ET] [EP] [LABL] [FFRM x] [EOF n] [FFRM 133] (FFRM 133) (EOF 3)
	PRINT	PTRp	Txxx [FILE n] [CONT] [DUMP] (all the tape) (readlabel) (no labels, no EOF)
			[SP1] [MAC] [SP2] [NOC] [END] [JOB jjjj] [SYST] [SP3] [ASA] (no eject) (no skipping) (syst/no syst) [SPCE] (SPC)
	PUNCH	PCHc	Txxx [DEN2] [EP] [DEN5] [ET] [94TP] [UFRM] [DEN8] [OP] [NLBL] [VFRM] [OC] [OT] (NLBL) [FFRM x] [EOF x] (FFRM 80) (EOF 3)
			[FILE x] [CONT] [COLB] [FILE 1] (readlabel) (EBCD)
	PUNCHSEP	RDRr PCHc	Txxx {same as for PUNCH} [END (no insert)]

	EJECT	PTRp		[P (6)]	[1 (0)]				
	TPCHECK	Txxx		[DEN2 DEN5 (DEN8)]	[EO OC OP (EP)]	[FILE x (FILE 1)]	[RCL (no RCL)]	[SENS (no sense listed)]	
				[DEN 2 DEN 5 (DEN 8)]	[TASK COPY TASK COMP (TASK BOTH)]	[IN xxxxxx (IN ----)]	[OUT yyyyyy (OUT ----)]		
(R)	TPCOPY	Txxx	Tyyy	[EOF n (all)]	[PAR ODD PAR EVEN (PAR BOTH)]	[CON ON (CON OFF)]	[TRAN ON (TRAN OFF)]		
				[MSGs OFF (MSGs ON)]	[TRYs n (TRYs 5)]	[BUFF s]			
	T7DUMP	PTRp	Txxx (7 tr)	[DEN 2 DEN 5 (DEN 8)]	[FILE n (FILE 1)]	[BIN BCD EVEN (mixed)]			
(R)	T9DUMP	PTRp	Txxx	[FILE n (FILE 1)]	[EBCD (Hex only)]				
	LABEL	Txxx		BINxxx	RELxxx	[DEN2 DEN5 (DEN8)]	[VOLx (VOL1)]	[ID (lincolnlab)]	
	CALCOMP	Txxx	PLTh	[Fn (no file skipping)]	[Ph (no picture skipping)]	[CONT (new tape)]			
(R)	UCSLOAD	RDRr	PTRp	[FOLD (no folding)]					
(R)	HNPLOAD		PTRp	[FOLD (no folding)]	[PN (HN)]				
	DISKDUMP	PTRp	Dddd	[cyl trk rec (VTOC)]	[maxcyl maxtrk (entire disk)]				
	SHARE	RDRr	PCHc	PTRp	Txxx	JOB n	[DEN2 DEN5 (DEN8)]		

Sense Switch Usage Summary

Jobs		REPRO	CONVERT	PRINT	PUNCH	PUNCHSEP	CALCOMP	SCOPE	Bit Function
Bits									
Standard Sense Switches	0			X	X	X			Terminate immediately
	1			X	X	X	X		Terminate at end of record
	2	X	X	X	X	X			Terminate at end of file
	3							X	Allow operator to TALK
	4								Reserved
	5								Reserved
	6								Reserved
	7			X	X	X			Pause immediately
Job Oriented Sense Switches	8							X	Delete unit check messages
	8			X					Single spacing
	9			X					Double spacing
	10			X					Triple spacing
	11			X					Program control
	12			X					Program control and eject
	13								Unused
	.								.
	.								.
	.								.
	31								Unused

Buffer Usage Summary

Buffers marked with an asterisk (*) can be changed using the CHANGBUF job.
Multiple buffers of the same size must all be changed to the same size.

<u>Job</u>	<u>Buffer No.</u>	<u>Buffer Size</u>
1. JOBS TASKS TERM SWITCHES }	none	
2. JOBDUMP LOWCORE COREDUMP }	none	
3. FSR, BSR, FSF, BSF, REW, RUN, WTM, ERG }	1 2	64 72
4. PRESTORE	1, 2, 3 4	168 120
5. PRINT	1, 2 3	2000 * 812
PUNCH	1, 2 3	200 * 812
PUNCHSEP	1, 2 3	200 * 812
EJECT	1	812
6. TPCHECK	1	256
7. TPCOPY	1, 2 3	2048 * 1024
8. T7DUMP	1 2 3 4	2000 * 168 36 132
9. T9DUMP	1 2	256 * 480
10. LABEL	1	204
11. CALCOMP	1	1772
12. CHANGBUF	none	

	<u>Job</u>	<u>Buffer No.</u>	<u>Buffer Size</u>
13.	DISKDUMP	1	3630
		2	7300
		3	96
14.	HNPLOAD	1	868
	UCSLOAD	1	868
15.	CONVERT	1	660
	REPRO	1	660
16.	PUNCHTAP	1	96
		2	200
		3	100
17.	LESTER	1, 2	960
		3	133
18.	LOAD	1	2048 *
		2	816
19.	UNLOAD	1	284
20.	SHARE	1	2048

B. Job Write-Ups

This section contains write-ups for each job in the system. The jobs are grouped into three categories: System Jobs, Dump Jobs, and Utility Jobs. Each write-up contains the following information:

- a) General information on the use of the job.
- b) Job summary.
- c) Task request format.
- d) Operating notes.
- e) Examples of the job request.
- f) Console messages.

Lincoln Laboratory Multi-Programming Supervisor

Job Operation Instructions

System Jobs

15 July 1966

General Information:

The system jobs do not require any devices except the use of the console typewriter. They should be included with all supervisor systems.

System Jobs: JOBNAME PARAMETERS

A. JOBS

Description:

Prints on the console typewriter a list of the jobs which can be initiated. The required devices for each job are indicated following the jobname; and a version code precedes the jobname. If a job is re-entrant, i. e. , can be activated for more than one task, the symbol (R) precedes the version code.

B. TASKS

Description:

Prints on the console typewriter a list of the jobs which are active, (i. e. , tasks) indicating the device units being used and the task number for each active job.

C. TERM tttt

Description:

Terminates an active task where tttt is the task number. The device units associated with the task are freed and become available for other tasks. This job is re-entrant and may be re-issued to terminate more than one task.

Console Message:

1. JOB NUMBER NOT IN JOB TABLES

The task number tttt specified in the request does not correspond to an active job. Either the job has already terminated or a task with this number has not been initiated.

D. SWITCHES tttt S_1 S_2 S_k

Description:

Displays, or modifies and displays, the sense switches of an active task where tttt is the task number and S_1 , S_2 , S_k are the sense switch bit numbers that are to be changed. The sense switches are located in the task's job table and occupy one word. They are numbered left to right from 0 to 31. The appearance of a sense switch bit among the task parameters causes that bit to be complemented (changed from 0 to 1 or from 1 to 0). The resulting values of the sense switches are then displayed. If no sense switch bit numbers are given, no changes are made to the sense switches, but they are displayed.

Console Messages:

1. PARAMETER ERROR

At least one of the sense switch bit parameters is outside the range of 0-31, or the task number tttt is zero.

2. TASK tttt jobname SWITCHES ARE

0: ssssssss 8: ssssssss 16: ssssssss 24: ssssssss

The switches for a specified task are displayed on the console typewriter where tttt is the task number, jobname is the corresponding job name, and ssssssss are the binary values of the sense switch bits (0 or 1).

Lincoln Laboratory Multi-Programming Supervisor

Job Operation Instructions

Dump Jobs

16 March 1966

General Information:

The dump jobs are available for use to aid in the debugging of a job program. The dump uses the BPS single phase dump routines which must be included as part of the supervisor. The dump appears on the system output unit (printer F on the Model 40 and the printer on the Model 65).

Dump Jobs:

JOBDUMP tttt

A dynamic dump of task tttt, where tttt is a task number of an active job, is produced on the system output unit. This job simulates an SVC JOBDMP (SVC 0) for the active job to dump the job program together with its buffers and the supervisor tables which contain status information of the job. The dump produced includes the JOB LIST, JOB TABLES, DEVICE LIST, DEVICE TABLES, and the ENTRY POOL, the CPUQ, and the first 4096 bytes of the job program as well as each job buffer. This job is re-entrant.

Console Message: JOB NOT ACTIVE

The task number, tttt, specified in this request does not correspond to an active job.

LOWCORE

A dump of the first 1000 (hex) bytes of core is produced on the system output unit. This area of core contains the I/O old PSW, CSW, CAW, and location of the device table at the time of the last 194 I/O interrupts. The location at which the last I/O old PSW is stored is indicated in the half word at location 56.

COREDUMP

A dump of core is produced on the system output unit. This coredump contains the LOWCORE information as well as a dump of the supervisor, all its tables, and all job programs. This job is re-entrant.

Lincoln Laboratory Multi-Programming Supervisor

Job Operation Instructions

Tape Control Jobs (Re-Entrant)

1 April 1966

General Information

The tape control jobs include the following:

FSR	Forward space n records
BSR	Backspace n records
FSF	Forward space n files
BSF	Backspace n files
REW	Rewind tape
RUN	Rewind and unload tape
WTM	Write n tape marks (end of files)
ERG	Erase n records gaps (3/4" of tape/gap)

Task Request Format: JOBNAME UNIT PARAMETERS

Jobname:

Any of the tape control job names listed above.

Unit: Txxx

Where xxx is the address of a 7-track or a 9-track tape unit.

Parameters:

Density:	DEN	2	(for 200 bpi)	Default:	DEN	8
		5	(for 556 bpi)			
		8	(for 800 bpi)			

The density code specifies the density of the tape on which the control operation is to be performed.

Count:	n	$0 \leq n \leq 9999$	Default:	1
--------	---	----------------------	----------	---

The count specifies the number of times the control operation should be performed.

The order of the parameters is immaterial, i. e. , the density parameter pair may come before or after the count. If a parameter is not specified, the default condition is used.

Operating Notes:

1. If the density specified by the operator is widely different from the true tape density, e. g. , if 800 is specified for a tape with a true density of 200, the files and record spacing operations may result in an EQUIPMENT MALFUNCTION error message.
2. If during a FSR or a BSR a tape mark is encountered, the operation is terminated and the remaining number of records not skipped are printed in a console message.
3. These jobs are re-entrant. Consequently, the same operation may be initiated for more than one tape unit and run concurrently.
4. The REW and RUN job does not require a density setting nor a count parameter. The ERG job uses the count parameter but does not require a density setting. All the other jobs use a density setting and a count parameter.

Examples:

1. FSR T180 DEN 5 300
Tape 180 will be advanced 300 records at a density of 556 bpi.
2. FSF T181 15 DEN 2
Tape 181 will be advanced 15 files at a density of 200 bpi.
3. BSF T181 5
Tape 181 will be backspaced 5 files at a density of 800 bpi.
4. WTM T180
One tape mark (EOF) will be written on Tape 180 at a density of 800 bpi.
5. REW T282
Tape 282 will be rewound to load point.

Console messages:

1. INCORRECT DENSITY.

The operator specified a density code that was not 2, 5, or 8. These are the only permissible codes.

2. COUNT IS NOT IN BOUNDS.

The count specified by the operator is less than zero or greater than 9999.

3. TAPE UNIT NOT READY

The tape unit is not ready. The operation will remain suspended until the unit is made ready. If the unit was in operation and became not ready, the operation will resume when the unit is made ready.

4. FILE MARK REACHED. REMAINING RECORDS: n

Tape mark reached during a BSR or a FSR operation. The remaining records (n) is the count specified by the operator minus the number of records skipped before the tape mark was encountered.

5. END OF TAPE
 REMAINING COUNT: n

The end of tape marker was reached during a FSR, FSF, WTM, or ERG operation. The remaining count (n) is the count specified by the operator minus the number of operations completed successfully before the end of tape marker was encountered.

6. LOAD POINT
 REMAINING COUNT: n

The load point marker was reached during a BSR or a BSF. The remaining count (n) is the count specified by the operator minus the number of records or files skipped.

7. EQUIPMENT MALFUNCTION. SENSE BYTES: xxxxxxxxxx

An equipment failure was detected. The sense bytes should be examined to determine the cause of failure. (See IBM Principles of Operation for 2400 series tape units. A22-6866-1.)

8. Txxx IS IN FILE PROTECT MODE. OPERATION NOT EXECUTED.

The tape unit Txxx was found to be in file protect mode when an attempt was made to execute a WTM or ERG job. The job was not executed.

PRESTORE

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Prestore Job

16 March 1966

Revised 27 May 1966

General Information:

PRESTORE performs the task of recording card images on magnetic tape. The program runs in two modes; for the 7094 it will prestore both BCD and binary card images adding the proper look-ahead bits to each record, for the 360 it will prestore EBCDIC card images in odd parity with the converter on. The program uses the TPWRUC, CDRDUC, and CDSTUC routines for tape and card unit checks.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: PRESTORE

Units: RDRr Txxx

Where r is the address of a 2540 card reader and xxx is the address of either a 7 or 9-track tape. If there is only one reader attached to the system, the reader unit is not specified.

Parameters:

Density:	DEN 2	Default: DEN 8
	DEN 5	
	DEN 8	

Mode:	BCD	Default: EBCD mode
-------	-----	--------------------

In the EBCD mode, 80 EBCDIC characters from each card are recorded on tape with the data converter on, and on an end-of-file condition on the card reader a tape mark is written.

In the BCD mode, each BCD card is recorded as an 84 character BCD record, with even parity, translator on; each binary card is recorded as a 168 character binary record with odd parity, translator off; each end-of-file card is recorded as a tape mark. Binary cards are indicated by a 7-9 punch in column 1 and end-of-file cards are indicated by a 7-8 punch in column 1. An end-of-file condition on the card reader does not cause a tape mark to be written.

Continuation: MORE

Default: Task Termination

When an end-of-file condition on the reader has been reached, this parameter causes the task to be suspended, i.e., remains active; otherwise, the task is terminated after the last card is written on tape.

Examples:

1. PRESTORE T181

Conventional prestoring for the 7094 will be started. The task terminates when the last card has been written and the end-of-file light is on.

2. PRESTORE T182 EBCD MORE

EBCDIC card images will be placed onto tape 182 with a tape mark written when the last card has been written and the end-of-file light is on. The task will remain active, and can be resumed by placing more cards into the hopper. In this mode, the task can be terminated only by the TERM task.

Console Messages († indicates task terminates)

1. CARD TO TAPE PRESTORE PROGRAM VERS 4.

This message is written when the task has begun successfully.

2. † PARAMETER ERROR

An erroneous parameter has been typed in, retry the task.

3. TAPE UNIT IS NOT READY.

This message indicates that the tape unit specified is not in the ready state at the start of the task.

4. JOB NO. xxxxx, nnnn CARDS. JOB NO. yyyyy BEING PRESTORED.

This message indicates that JOB xxxxx has been successfully prestored and contained nnnn cards. JOB yyyyy is being started.

5. TAPE PARITY

This message indicates that there was difficulty in writing on the tape. If this message is repeated many times a new tape should be mounted.

6. † END OF TAPE REEL

The end of tape indicator has been reached. Mount a new tape and begin with the last job to be prestored.

7. † nnnnn CARDS PRESTORED

This is a count of the total number of cards prestored for the task.

8. CARD 2 IN STACKER HAS INVALID PUNCH IN COLUMN xx.

This message indicates that a BCD card has an invalid punch in column xx. The next to last card in stacker 2 should be inspected in the specified column and the condition should be corrected. This card and the one following, together with the cards to be run out, should be refed.

Note: No error light on the reader will be lit for this condition. If one of the lights does come on CDRDUC will print a message and only those cards that are run out should be refed.

PRINT/PUNCH
PUNCHSEP/EJECT

Lincoln Laboratory Multi-Programming Supervisor
Job Operating Instructions
PRINT/PUNCH/PUNCHSEP/EJECT Jobs (Re-entrant)

1 April 1967

Job Summary:

PRINT processes labeled and unlabeled 7 or 9-track tapes. 7-track tapes may be read under any density and mode set specification. 9-track tapes force 800 bpi and odd parity. Mixed density tapes cannot be printed.

PUNCH punches a 7-track tape written in odd and/or even parity or a 9-track tape. Cards can be punched as column binary, BCD, or EBCDIC. Tapes may be labeled or unlabeled and may be of any density.

PUNCHSEP is like the PUNCH job except that two blank cards from the reader are inserted into the punched deck whenever an end of file or system output flip card is recorded on the tape.

EJECT restores and/or spaces the paper on a printer.

General Information:

TYPE OF TAPES PROCESSED. PRINT, PUNCH and PUNCHSEP process 360 labeled and unlabeled 7 or 9 track tapes and 7094 unlabeled 7-track tapes. Seven track tapes may be read under any density and mode-set specification. Nine track tapes force 800 bpi and odd parity. Mixed density tapes cannot be processed.

PROCESSING OF LABELED TAPES. When processing labeled tapes, only 80 byte records with VOL, HDR, UHL, EOF, UTL or EOVS in bytes 1-3 are accepted as standard labels. If a non-standard label is read the program assumes that the tape is positioned erroneously within the data set. The tape is then backspaced two files to position if ahead of the header labels, a message is typed on the console typewriter, and the processing of labels is resumed.

SKIPPING TO nth DATA SET ON LABELED TAPE. To skip to a specific data set on a labeled tape the forward space file job, FSF, must be used to position the tape at the header labels preceding the data set desired. The number of files to skip is determined by the formula $3n-3$ where n is the ordinal number of the data set. (To position the tape at the beginning of the data set use the formula $3n-2$.) The tape must be at load point before initiating the FSF job.

SYSTEM OUTPUT TAPES. Labeled and unlabeled system output tapes are generated under the Operating System/360 and unlabeled system output tapes created under the IBSYS System on the 7094 are supported by the PRINT, PUNCH and PUNCHSEP jobs. When an OS/360 job statement card image or an IBSYS "\$ID" card image is encountered, the PRINT job writes the inventory number appearing in each of these records on the console typewriter. Also, four rows of asterisks at the top and/or bottom of three pages are printed to facilitate the separating of user job listings. When a 360 tape is being punched, 161 byte records with hexadecimal "A1" or "A2" as the first byte of the record are recognized as flip card images and are punched as column binary. If the PUNCHSEP job is in effect and if the flip card record has "A1" as the first byte, two cards from the reader precede the flip card into the punched deck. When a 7094 tape is being punched, odd parity records of 84 bytes or less with "\$JOB" in bytes 1-4 are recognized as 7094 FORTRAN IV job statement card images. The record is punched as BCD and if PUNCHSEP is in effect, two blank cards from the reader are inserted into the deck.

RECORD FORMATS. OS/360 labeled and unlabeled tapes may have fixed, variable and unspecified record formats. Unlabeled 7094 print tapes may be unblocked or blocked with record marks as logical record separators. Unlabeled 7094 punch tapes may be blocked or unblocked for odd parity and should be unblocked for even parity. For odd parity records, 160 bytes are punched per card in column binary mode and every 169th byte is treated as the first byte of an additional logical record with the previous eight bytes ignored. For even parity records 80 bytes are punched per card in EBCD mode with any excess bytes ignored.

CONTROL CHARACTERS. Carriage control characters in either extended ASA code or machine command code are recognized by the PRINT program. Records with no carriage control characters are printed single spaced. For machine code carriage control both write command and independent carriage select commands are permitted. The two command types are executed for the printer except that records containing independent carriage select commands are not printed. The following ASA codes are processed:

<u>CHAR</u>	<u>HEX CODE</u>	<u>RESULT</u>
BLANK ()	'40'	Space 1 line immediately
ZERO (0)	'F0'	Space 2 lines immediately
MINUS (-)	'60'	Space 3 lines immediately
PLUS (+)	'4E'	No-op command; used for overprinting
NUMBERS (1-9)	'F1' - 'F9'	Skip to channel 'N' immediately

All other characters are overridden by space 1 line immediately.

MAXIMUM PHYSICAL/LOGICAL RECORD LENGTH. The maximum physical record length is 1998 bytes for PRINT and 198 for PUNCH and PUNCHSEP. This maximum may be changed temporarily by using the CHANGBUF program to alter the size of buffer one and two before execution. Upon initiation, the program resets buffer sizes to 2000 (or 200) for subsequent jobs (buffer sizes must be two more than the maximum physical length). For PRINT, the maximum logical record length is 132 bytes or 133 bytes if program control is in effect. For PUNCH and PUNCHSEP, the maximum is 80 bytes for EBCDIC punching and 160 bytes for column binary punching. If a logical record exceeds the maximum, 132 bytes (80 or 160) are printed per line (or punched per card) until the logical record is exhausted.

JOB TERMINATION. Unless otherwise specified by the EOF parameter, a task is terminated when three successive tape marks (end of files) are sensed on the tape. When processing labeled tapes a task is terminated when two successive end of files are sensed following a trailer label, or the words "END OF OUTPUT" appear in bytes 2-14. If a job is terminated because of "END OF OUTPUT" the remaining data set is not printed. Also, if the DUMP parameter is specified, the remaining trailer labels and end of file messages are not printed. If the FILE parameter is specified, a task is terminated when the request is satisfied. Also, a tape mark in the end of tape area causes job termination.

SYSTEM ROUTINES. Use is made of the standard system unit check and unit exception routines, TPRDUC, PRTUC, PRTUE, CDRDUC, CDPNUC, CDSTUC and CDPNUE. The SWITCHES system job may be invoked to control PRINT, PUNCH and PUNCHSEP. The terminations accomplished by setting switch bits one and two are "clean", i.e., they allow the job to be re-initiated without producing any different results than if the task had been allowed to continue. If a task which involves a labeled tape is terminated by switch setting 2, the tape is positioned at or within the data set. Before terminating, the task writes a message on the console typewriter indicating the parameters which must be listed in order to resume the task properly. Bit seven causes a task to suspend immediately; the task can be continued by resetting bit seven. Once a spacing sense switch bit has been set it will remain in effect until it is reset and a new spacing switch bit is selected.

Task Request Summary:

	<div>[DEN2] DEN5 (DEN8)]</div>	<div>[OP] OT OC EP (ET)]</div>	<div>[94TP] NLBL (LABL)]</div>	<div>[UFRM VFRM FFRM x (FFRM 133)]</div>	<div>[EOF n] (EOF 3)]</div>
PRINT	PTRp	Txxx	<div>[FILE n (all the tape)]</div>	<div>[CONT (read label)]</div>	<div>[DUMP (no labels, no EOF)]</div>
	<div>[SP1 SP2 SP3 SPCE (SPC)]</div>	<div>[MAC NOC (ASA)]</div>	<div>[END (no eject)]</div>	<div>[JOB jjjj (no skipping)]</div>	<div>[SYST (syst/no syst)]</div>

Sense Switch Bits Used:

- | | |
|--------------------------------|--------------------------------|
| 0 - terminate immediately | 8 - single spacing |
| 1 - terminate at end of record | 9 - double spacing |
| 2 - terminate at end of file | 10 - triple spacing |
| 7 - pause immediately | 11 - program control |
| | 12 - program control and eject |

	<div>[DEN2] DEN5 (DEN8)]</div>	<div>[EP ET OP OT (OC)]</div>	<div>[94TP LABL (NLBL)]</div>	<div>[UFRM VFRM FFRM x (FFRM 80)]</div>	<div>[EOF x (EOF 3)]</div>
PUNCH	PCHc	Txxx	<hr/>		
	<div>[FILE x (FILE 1)]</div>	<div>[CONT (read label)]</div>	<div>[COLB (EBCD)]</div>	<hr/>	

Sense Switch Bits Used:

- 0 - terminate immediately
- 1 - terminate at end of record
- 2 - terminate at end of file
- 7 - pause immediately

PUNCHSEP	RDRr	PCHc	Txxx	{ same as for PUNCH }	[END (no insert)]
----------	------	------	------	--------------------------	------------------------

Sense Switch Bits Used:

- 0 - terminate immediately
 - 1 - terminate at end of record
 - 2 - terminate at end of file
 - 7 - pause immediately
-

EJECT	PTRp	[p] [(6)]	[1] [(0)]
-------	------	------------------	------------------

Sense Switch Bits Used: None

Task Request Formats: JOBNAME UNITS PARAMETERS

Jobname: PRINT

Units: PTRp Txxx

where p is a printer address and xxx is the address of a 7 or 9-track tape unit.

Jobname: PUNCH

Units: PCHc Txxx

where c is a 2540 card punch and xxx is a 7 or 9 track tape unit address.

Jobname: PUNCHSEP

Units: RDRr PCHc Txxx

where r is a 2540 card reader, c is a 2540 card punch and xxx is a 7 or 9 track tape unit address. The reader and punch must be on the same 2540 card unit. The program does not check for this requirement.

Jobname: EJECT

Units: PTRp

where p is a printer address.

Parameters: For the PRINT, PUNCH, PUNCHSEP jobs

Density: Default: DEN8

DEN2 (200 bpi)

DEN5 (556 bpi)

DEN8 (800 bpi)

Mode set: PRINT Default: ET
PUNCH, PUNCHSEP Default: OC

OP (odd parity - converter off - translator off)

OT (odd parity - converter off - translator on)

OC (odd parity - converter on - translator off)

EP (even parity - converter off - translator off)

ET (even parity - converter off - translator on)

For labeled tapes, the mode set option is ineffective. All labels are read with ET and all data sets are read according to label specifications.

Tape Type: PRINT Default: LABL
PUNCH, PUNCHSEP Default: NLBL

LABL (360 labeled tape)

NLBL (360 unlabeled tape)

94TP (7094 tape)

IBM 7094 print tapes are read with converter off, translator on with even parity. Records are treated as though written in unspecified record format except that record marks, if any, are interpreted as logical record delineators and are not printed. Also, hexadecimal "50" is translated to hexadecimal "4E" due to the print chain arrangement. 7094 punch tapes may be written in odd and/or even parity. Odd parity records are read with converter off, translator off and are punched as column binary. Even parity records are read with converter off, translator on and are punched as EBCD.

Record format: PRINT Default: FFRM 133
PUNCH, PUNCHSEP Default: FFRM 80

FFRM xxx (fixed record format)

VFRM (variable record format)

UFRM (unspecified record format)

where xxx is the logical record length which may be any number from 0 to 133. If xxx is 0, the logical record length equals the physical record length. The logical record length must be specified for fixed record format only. With labeled tapes the record format parameters are ineffective because they are overridden by the label specifications.

Read data:

Default: read label

CONT

This parameter is used when processing a labeled tape which initially is positioned within a data set. Because the job program cannot obtain label information concerning the mode and record format in which the data set is written without backspacing the tape two files, a mode set parameter and a record format parameter must accompany the CONT parameter. If they do not, the "parameter error" message is written on the console typewriter and the job is terminated. A mode set parameter must also be specified when reading 9 track tapes but it is ineffective. The CONT parameter is usually used in resuming a job which was prematurely terminated by setting sense switch 2.

Terminate on x tape marks:

Default: EOF 3

EOF x

where x may be any number from 1 through 9. This parameter causes the job to terminate after x successive tape marks (EOF) are sensed on the tape.

Terminate after x files:

PRINT Default: all the tape

PUNCH, PUNCHSEP Default: FILE 1

FILE xxxx

where xxxx may be any number from 1-9999. The file parameter is used when it is desired to terminate the job after printing xxxx number of files.

Depending on the type of tape being printed "FILE" has different meanings.

- 1) For unlabeled tapes a file consists of one or more records terminated by a tape mark.
- 2) For IBSYS system output print tapes, a file consists of a job. An IBSYS job may or may not begin with a \$ID card image and is terminated by either a tape mark or the \$ID card image of the following job.

- 3) For OS/360 user labeled output tapes, a file refers to a single data set.
- 4) For OS/360 SYSOUT print tapes, a file consists of all data sets belonging to a job. Each job is identified by the job statement card.

Start location:

Default: Print without
skipping

JOB xxxxx

where xxxxx may be any number from 1-99999. The program will skip to and begin printing from the job identified by inventory number xxxxx. If the tape is an OS/360 SYSOUT tape the number xxxxx appears in bytes 4-8 and the word "JOB" in bytes 9-11. If the tape is an IBSYS SYSOUT tape the number xxxxx appears in the "\$ID" card image with the number xxxxx and "\$ID" located in one of 3 positions within the record.

	\$ID	NUMBER
bytes	2-4	51-55
bytes	7-9	56-60
bytes	13-15	62-66

The inventory number xxxxx consists of five digits, with leading zeroes if necessary. Only the significant digits need be specified in the parameter.

SYSOUT tape:

Default: syst/no syst

SYST

This parameter indicates that a print tape is an OS/360 SYSOUT tape. If the parameter is not specified, PRINT will determine if the tape is a sysout tape by searching for "SYSOUT" in bytes 5-10 of data set trailer label number one. This parameter need only be used with the CONT parameter.

Additional Parameters: For the PUNCH and PUNCHSEP jobs

The following parameter applies only to the PUNCH and PUNCHSEP jobs.

Punching data mode:

Default: EBCD

EBCD (data mode 1: 1 byte/card col.)

COLB (data mode 2: column binary, 2 bytes/card col.)

Additional Parameters: For the PUNCHSEP job

The following parameter applies to PUNCHSEP only.

Card image tape: END

Default: Blank cards
not inserted

The parameter causes two blank cards from the reader to be fed and inserted after an "END" card (" END" in columns 9-12) is punched. The parameter is used when punching tapes on which machine language source decks are prestored.

Additional Parameters: For the EJECT job

The following positional parameters apply to EJECT only.

No. of pages to eject: Default: 6

p

where p may be any number from 0-10.

No. of lines to space: Default: 0

l

where l may be any number from 0-66.

Operating Instructions:

1. Whenever a print job terminates, except on premature termination caused by setting sense switch 2, 6 pages are ejected on the printer.
2. All punched cards are directed to stacker RP3. There is no option available for changing stacker designation. Badly punched cards are stacked in stacker P1. Before terminating, a blank card is punched into stacker P1 to release the last card punched from the tape into stacker RP3. Before inserting a blank card from the reader into the punched deck when an end of file is sensed on the tape, PUNCHSEP also punches a blank card into stacker P1 to release the last card into stacker RP3.
3. When PUNCHSEP is used, cards in the reader and punch hoppers should be of contrasting colors.

Examples:

1. PRINT PTRE T183

The tape on unit 183 is assumed to be an OS/360 labeled tape of density 800. Labels are read in even parity, translator on and data sets are read and printed according to label specifications. The tape is printed on printer E under program control and the job terminates when three successive end of files are sensed on the tape, when two end of files are sensed following a trailer label, or the words "END OF OUTPUT" appear in bytes 2-14. (9-track tapes force 800 bpi and odd parity.

2. PRINT PTRF T280 JOB 437 FILE 1 DUMP SP1

Tape on unit 280 is read as a 360 labeled tape and searched for job 437. If found, one file (i. e., one job) is printed single spaced on printer F. All labels are printed and all end of files are indicated on the listing.

3. PUNCH PCHB T182

Tape on unit 183 is read at 800 bpi, converter on, translator off (7-track only). The tape is assumed to be unlabeled with records in fixed record format with 80 bytes per logical record. The tape is punched on punch B and the job terminates when a tape mark is sensed, i.e., one file is completed.

4. PUNCHSEP RDRC PCHD T181 94TP FILE 100

Tape on unit 181 is punched as a 7094 tape. Odd parity records are punched as column binary; even parity records are punched as EBCD. If any flip cards are found or tape marks are sensed two blank cards from the reader are inserted into the punched deck. The job terminates after processing 100 files or upon sensing 3 successive end of files.

5. EJECT PTRF

6 pages are ejected on printer F.

6. EJECT PTRF 0 33

The paper on printer F is spaced 33 lines.

Console Messages: For the PRINT, PUNCH and PUNCHSEP jobs

(† indicates task terminates)

All console messages are preceded by the printer or punch unit ID and the tape unit ID. For example:

--PTRE T181--END OF TAPE
--PCHB T181--END OF TAPE

If only one printer or punch is attached to the system, the ID is not typed. For example:

--T181--END OF TAPE

1. † PARAMETER ERROR

An improper parameter was specified.

2. † ENLARGE BUFFERS 1 AND 2

Typed whenever a tape record exceeds input buffer size. Buffer sizes can be enlarged by using the CHANGBUF program. Buffers 1 and 2 must be of equal size.

3. † CANNOT ALTER SIZE OF BUFFER 3--REINITIATE JOB
Buffer 3 is a DSECT area and should not be changed by the CHANGBUF program.
4. † END OF TAPE
A tape mark in the end of tape area was sensed.
5. † MACHINE ERROR ON BACKSPACE UC
Invalid unit check indication was received while attempting to backspace tape.
6. † END OF JOB
Typed whenever the print job is terminated by the SWITCHES job.
7. † END OF OUTPUT
Typed whenever two end of files are sensed following a trailer label or the words "END OF OUTPUT" are recognized in columns 2-14.
8. † xxx FILES COMPLETED
Typed when a request invoked through the file parameter has been satisfied.
The number xxx is the number specified in the file parameter.
9. † x END OF FILES
x successive end of files were sensed on tape. If the EOF parameter is not used, the job is terminated on 3 end of files (EOF 3).
10. † TO REINITIATE, MUST SPECIFY: CONT mode set format (system) (control)
The above message is typed when a job which involves a labeled tape is prematurely terminated by setting sense switch 2. The parameters listed in the message are the parameters which must be specified when resuming the job. The SYSTEM parameter is specified in the message only if the print tape is a SYSOUT tape. A carriage control parameter is specified only if the data set has MAC or NOC carriage control. Any parameters which were specified in the initial job request and which do not appear in the message must also be specified when resuming the job, e.g., the spacing, JOB, or EOF parameters.
11. TAPE PARITY
When a parity error is detected while reading, the record is reread 10 times. If the error still exists the record is accepted as read and the above message is typed. If the record in error is to be printed the following message appears in the listing:
****PARITY ERROR IN READING NEXT RECORD AFTER 10 RETRIES****

12. TAPE POSITIONED INCORRECTLY. BACKSPACED 2 FILES

Typed whenever a non-standard label is found while processing labeled tapes.

Additional Console Messages: For the PRINT job

1. REPEATING PRINT LINES

The PRINT program checks for consecutively repeating print lines. If a line is repeated more than 65 times the above message appears on the console typewriter and the remaining repeating lines are not printed. A count of how many times the line is repeated is obtained and when the repetitions are through the following messages are printed on the listing:

+++++++ABOVE LINE REPEATED xxxxxxxxxxTIMES. ONLY 66 LINES PRINTED.+++++++

where xxxxxxxxxx is the number of repetitions. Thereafter, the job resumes regular printing.

2. † MAXIMUM NUMBER OF LINES REPEATED. JOB TERMINATED.

A print line has been repeated hexadecimal FFFFFFFF times.

3. JOB xxxxx STARTED

Whenever a new job is started, its inventory number xxxxx as found in the OS/360 job statement card image or the IBSYS \$ID card image is typed.

Additional Console Messages: For the PUNCH job

1. FORTRAN IV FLIP CARD. SHOULD REQUEST PUNCHSEP JOB.
TO CONTINUE, STOP AND START PUNCH.

PUNCHSEP should be used when punching 7094 SYSOUT tapes. When PUNCHSEP is used, punching will remain a continuous operation by having blank cards read from the reader delineate job decks.

Additional Console Messages: For the PUNCHSEP job

1. READER NOT READY

The last card has been read in the reader.

2. CARDS IN READER NOT BLANK

A non-process run-out on the reader should be performed, the hopper filled with blank cards and the reader made ready.

3. † CARD STACK UE

A unit exception on a feed and select stacker operation has occurred. This indicates machine error and the CE should be notified.

Additional Console Messages: For the EJECT job

1. † PARAMETER ERROR

A faulty parameter was specified.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

TPCHECK Job (Re-entrant)

1 April 1967

Job Summary:

The TPCHECK job examines a 7-track or a 9-track tape for parity errors and converter checks, lists sense information on a permanent read error, switches parity from even to odd (or odd to even) if necessary, recording the location and number of switches, and determines the number of bytes per record and records per file.

General Information:

The TPCHECK job lists on the console typewriter

1. the location (file and record) of parity errors and converter checks.
2. the number of records per file.
3. the number of records with parity errors and converter checks per file.
4. the number of switches to odd or even parity per file.
5. optionally, the number of bytes per record.
6. optionally, the sense information on a permanent read error.
7. optionally, the location (file and record) of parity switches.

The channel command word (CCW) for reading the tape has the skip bit (bit 35) set to suppress the transfer of data from the channel to main storage. The system TPRDUC routine is used. Unless the mode set parameter EO is specified, no attempt is made at rereading the tape when a parity error occurs.

Task Request Summary:

TPCHECK	Txxx	$\begin{bmatrix} \text{DEN2} \\ \text{DEN5} \\ \text{(DEN8)} \end{bmatrix}$	$\begin{bmatrix} \text{EO} \\ \text{OC} \\ \text{OP} \\ \text{(EP)} \end{bmatrix}$	$\begin{bmatrix} \text{FILE x} \\ \text{(FILE 1)} \end{bmatrix}$	$\begin{bmatrix} \text{RCL} \\ \text{(no RCL)} \end{bmatrix}$	$\begin{bmatrix} \text{SENS} \\ \text{(no sense listed)} \end{bmatrix}$
---------	------	---	--	--	---	---

Task Request Format:

Jobname: TPCHECK

Unit: Txxx

where xxx is a tape unit address.

Parameters:

Density:	DEN2	(200 bpi)	Default: DEN8
	DEN5	(556 bpi)	
	DEN8	(800 bpi)	

Mode Set:		Default: EP
EP	(even parity, converter off, translator off)	
OP	(odd parity, converter off, translator off)	
OC	(odd parity, converter on, translator off)	
EO	(EP or OP, whichever is successful)	

The OC mode set is included so that converter check errors may be detected.

The EO parameter is used in reading a mixed mode tape or a tape whose parity is unknown. When a permanent read error occurs, the EO parameter causes the record in error to be reread with the mode set switched from EP to OP or OP to EP, depending on which mode set is currently in effect. If the reread is successful, the location (file and record) of the parity switch is indicated on the console listing and the job continues reading the tape in the new mode. If the retry attempt in the new mode proves unsuccessful, the record is reread in the original mode. If the error persists, the record is accepted as read and the "parity error" message appears on the console typewriter. The job continues reading the tape in the original mode. Initially, the tape is read with the mode set to EP.

No. of files: Default: FILE 1

FILE x

where x may be any number from 1 through 99999.

Record length listed: Default: no record
length listed

RCL

When the RCL parameter is specified, the number of bytes in each record is printed on the console typewriter. The size of the largest record which TPCHECK will process is 65,534 bytes. If the length of a record exceeds the maximum, a message will appear at the console and the job will continue with the next record.

Sense listed on parity error:
SENS

Default: no listing of
sense information

If a permanent read error occurs the sense information is written in hexadecimal on the console typewriter. All six sense bytes are listed. For a description of the sense bytes, refer to the IBM 2401, 2402, 2403, 2404, and 2816 Model 1 Principles of Operation Manual, A22-6866-2.

Examples:

1. TPCHECK T181

One file is read and the ordinal numbers of any record in which a parity error occurs is listed. Also, the total number of records in the file and the total number of records with parity errors, converter checks, and parity switches in the files, are listed.

2. TPCHECK T181 DEN5 OP FILE 3 RCL

Same as in example 1 except that 3 files are read at 556 bpi with odd parity, converter off, translator off, and the number of bytes in each record is listed.

3. TPCHECK T181 EO SENS

Same as in example 1 except that the sense information is listed on the console typewriter whenever a permanent read error occurs. Also, the tape is read with EP or OP whichever is successful. If neither mode set is successful, the record is accepted as a permanent error record.

Console Messages: († Indicates task terminates)

1. END OF FILE xxxxx, NO. RCS. = yyyyy, NO. PARITY ERRS = zzzzz,
NO. CONV CKS = vvvvv, NO. PARITY SWTS = wwwww.

At the end of each file, the total number of records in the file, and the number of records with parity errors, converter checks, and parity switches in the file are listed.

2. RCL. = bbbbb, FILE NO. = xxxxx, RC. NO. = yyyyy.

If the RCL parameter is specified, this message is printed after each record is read to indicate the number of bytes in the record.

3. RCL. EXCEEDS 65,534, FILE NO. = xxxxx, RC. NO. = yyyyy.

The above message is printed if the RCL parameter is specified and the length exceeds 65,534.

4. CONVERTER CHECK, FILE NO. = xxxxxx, RC. NO. = yyyyy.

Whenever a converter check is detected, the above message is typed.

5. PARITY ERROR, FILE NO. = xxxxxx, RC. NO. = yyyyy.

This message is printed whenever a parity error is detected in a record and the SENS parameter is not specified.

6. PARITY ERROR, FILE NO. = xxxxxx, RC. NO. = yyyyy.
SENSE = xxxxxxxxxxxxxx.

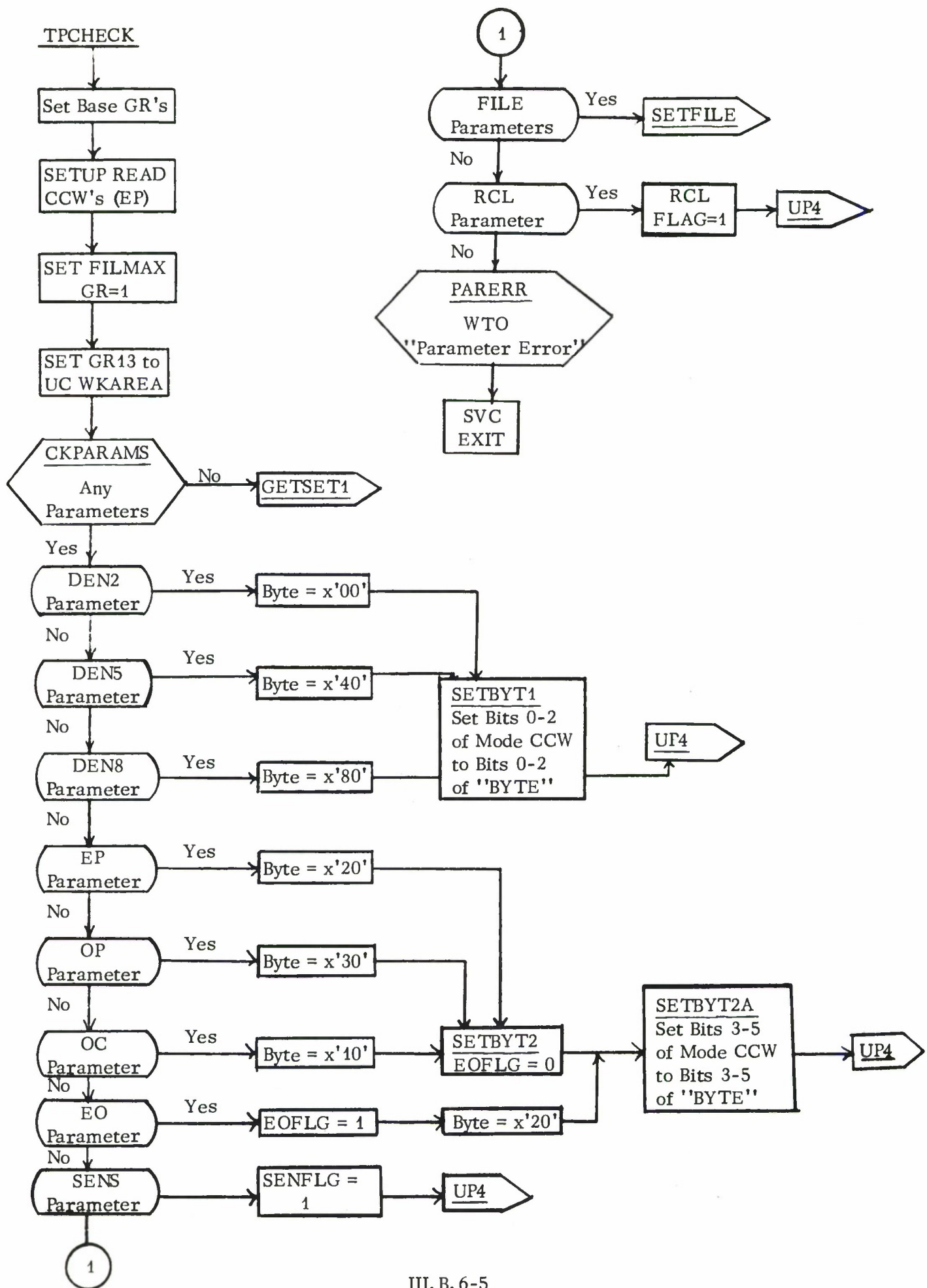
The above message is typed when a permanent read error occurs and the SENS parameter has been specified.

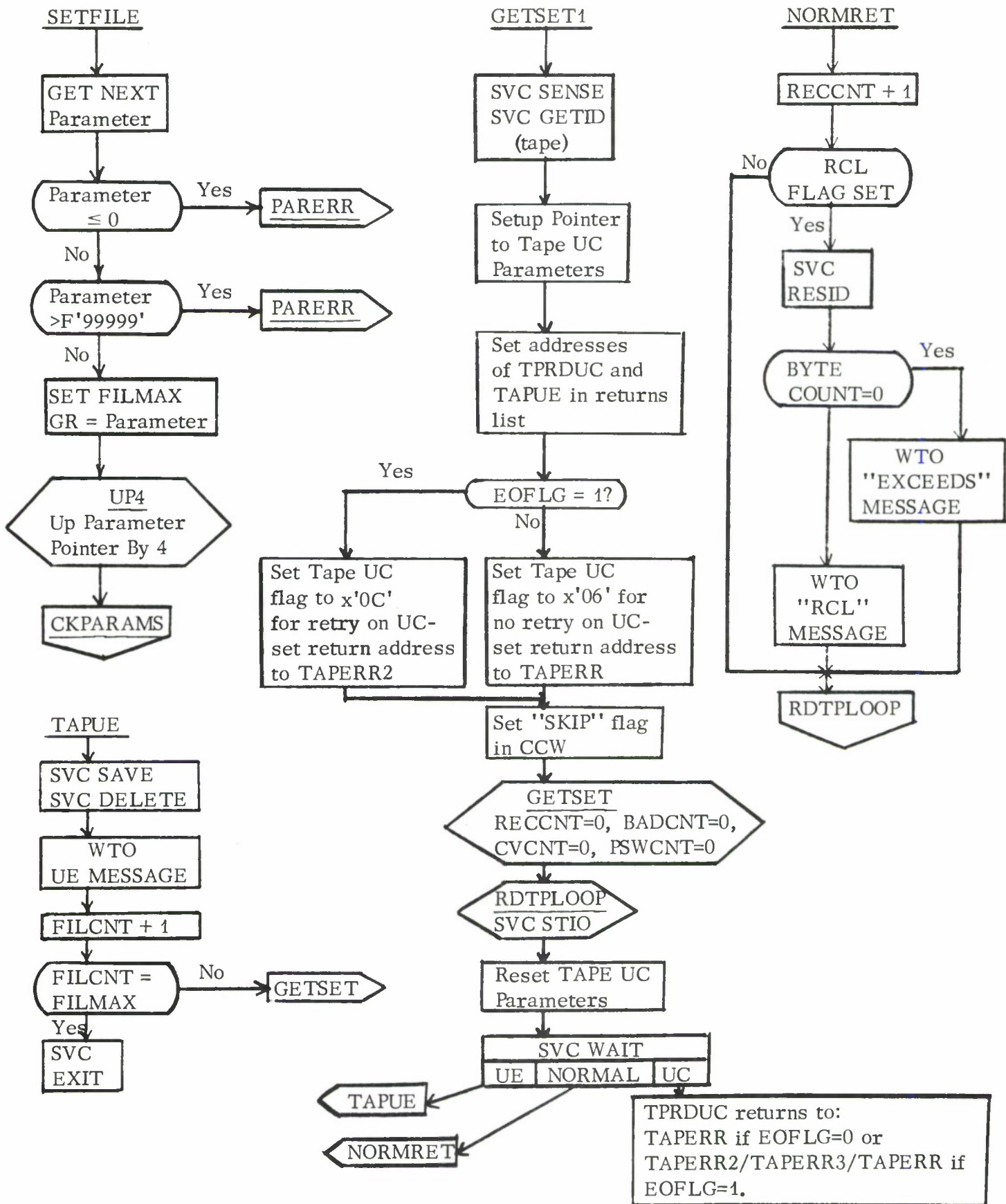
7. PARITY SWITCHED TO xx, FILE NO. = yyyyy, RC. NO. = zzzzz.

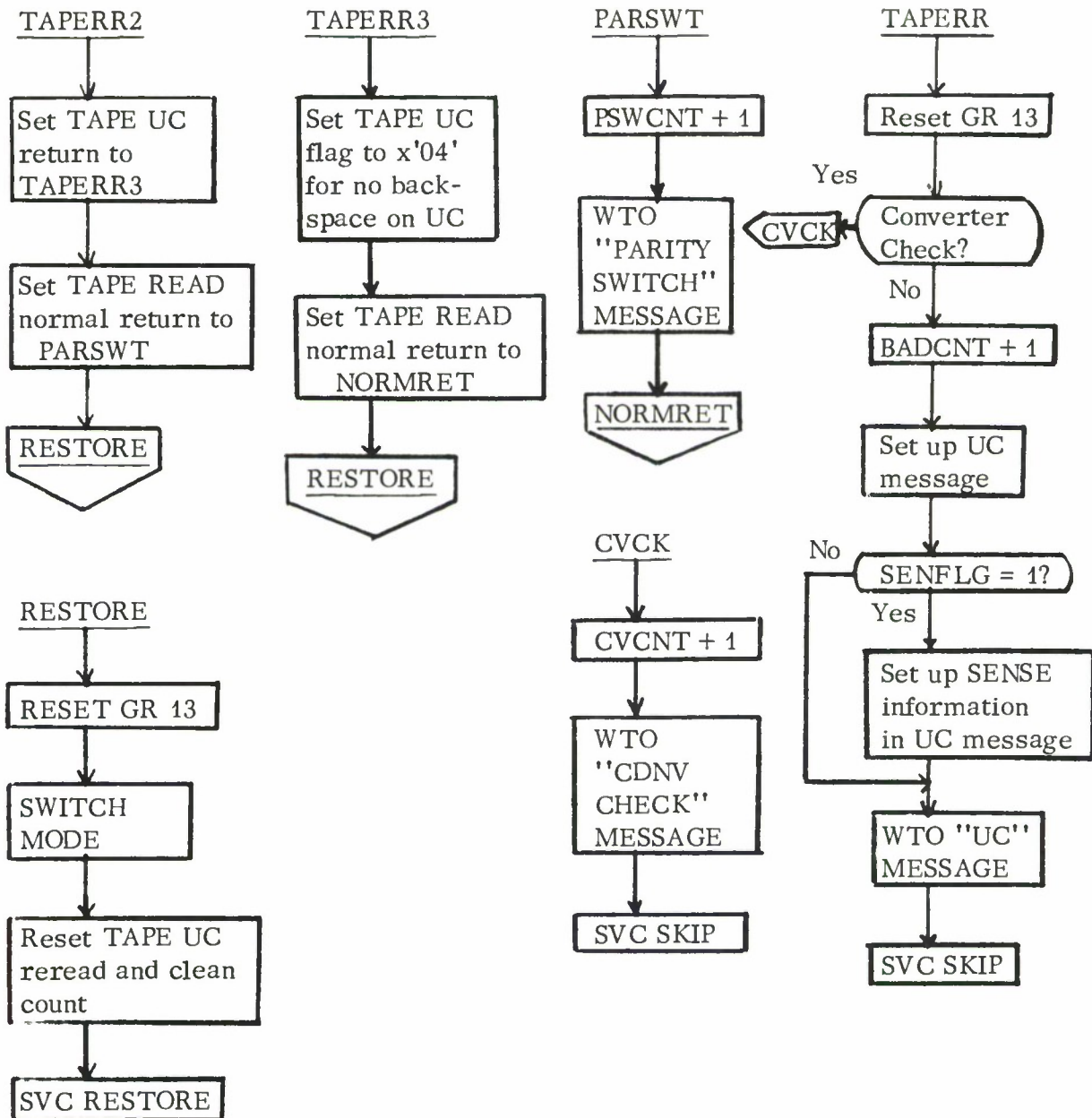
If the EO parameter is specified, the above message is typed at the console whenever the parity is switched. The field xx is either EP or OP, to indicate the parity in which the tape was last read.

8. † PARAMETER ERROR.

When an invalid parameter is typed the above message is listed and the job is terminated.







Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Tape Copy Job (TPCOPY)

(Re-entrant)

1 April 1967

Job Summary:

TPCOPY performs copy and/or comparison of 7- or 9-track tapes under the Multi-Programming Supervisor. Options are available for selecting parity, density, number of files, and use of the data conversion feature.

General Information:

Tapes should be positioned and readied before task initiation. Rewinding of units is only done by the program between the copy and compare phases of a task that calls for both phases. Thus, copying or comparing of individual portions of a tape (a number of files) to or from different reels may be performed as successive tasks without losing tape position.

Performing a copy or compare on tapes which do not provide multiple tape marks as an end-of-data indication (e.g., most PRESS tapes) may result in trying to read blank tape or an old recording. For this reason, it is desirable to specify the number of files when initiating a task.

Task Request Summary:

TPCOPY	Txxx	Tyyy	<table border="0"><tr><td>[DEN 2]</td></tr><tr><td>DEN 5</td></tr><tr><td>[DEN 8]</td></tr></table>	[DEN 2]	DEN 5	[DEN 8]	<table border="0"><tr><td>[TASK COPY]</td></tr><tr><td>TASK COMP</td></tr><tr><td>(TASK BOTH)</td></tr></table>	[TASK COPY]	TASK COMP	(TASK BOTH)	<table border="0"><tr><td>[IN xxxxx]</td></tr><tr><td>[IN -----]</td></tr></table>	[IN xxxxx]	[IN -----]	<table border="0"><tr><td>[OUT yyyyy]</td></tr><tr><td>[OUT -----]</td></tr></table>	[OUT yyyyy]	[OUT -----]	<table border="0"><tr><td>[EOF n]</td></tr><tr><td>[EOF all]</td></tr></table>	[EOF n]	[EOF all]
[DEN 2]																			
DEN 5																			
[DEN 8]																			
[TASK COPY]																			
TASK COMP																			
(TASK BOTH)																			
[IN xxxxx]																			
[IN -----]																			
[OUT yyyyy]																			
[OUT -----]																			
[EOF n]																			
[EOF all]																			
			<table border="0"><tr><td>[PAR ODD]</td></tr><tr><td>PAR EVEN</td></tr><tr><td>(PAR BOTH)</td></tr></table>	[PAR ODD]	PAR EVEN	(PAR BOTH)	<table border="0"><tr><td>[TRAN ON]</td></tr><tr><td>(TRAN OFF)</td></tr></table>	[TRAN ON]	(TRAN OFF)	<table border="0"><tr><td>[CON ON]</td></tr><tr><td>(CON OFF)</td></tr></table>	[CON ON]	(CON OFF)							
[PAR ODD]																			
PAR EVEN																			
(PAR BOTH)																			
[TRAN ON]																			
(TRAN OFF)																			
[CON ON]																			
(CON OFF)																			
			<table border="0"><tr><td>[MSG OFF]</td></tr><tr><td>(MSG ON)</td></tr></table>	[MSG OFF]	(MSG ON)	<table border="0"><tr><td>[TRY n]</td></tr><tr><td>(TRY 5)</td></tr></table>	[TRY n]	(TRY 5)	<table border="0"><tr><td>[BUFF s]</td></tr><tr><td>(BUFF 2048)</td></tr></table>	[BUFF s]	(BUFF 2048)								
[MSG OFF]																			
(MSG ON)																			
[TRY n]																			
(TRY 5)																			
[BUFF s]																			
(BUFF 2048)																			

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: TPCOPY

Units: Txxx Tyyy

Where xxx is the input tape unit number and yyy is the output tape unit number. The tape units should be made ready before task initiation.

Parameters: Any number of parameter pairs (name value) from the list below may be specified. Parameter pairs may be in any sequence; and, if two or more parameter pairs contain the same name, the value given in the last pair is used. The following parameters have precedence:

1. Requesting TRANslator ON overrides any converter specification regardless of sequence and results in obtaining CONverter Off. In this case, the current program also forces even parity.

2. Requesting CONverter ON overrides any parity specification regardless of sequence and results in obtaining odd parity.

If a parameter pair is not specified, the default value is assumed.

	<u>Parameter</u>	<u>Name</u>	<u>Values</u>	<u>Default Value</u>
1.	Density	DEN	2 (200 bpi) 5 (556 bpi) 8 (800 bpi)	8

Copying from 7-track to 7-track and specifying a density parameter value causes the input tape to be read and the output tape to be written with the requested density.

Copying from 7-track to 9-track and specifying the density parameter results in reading the 7-track tape with the requested density and writing the 9-track tape with 800 bpi.

Copying from 9-track to 7-track and specifying the density parameter results in reading the 9-track tape at 800 bpi and writing the 7-track tape with the requested density.

Copying from 9-track to 9-track forces 800 bpi.

	<u>Parameter</u>	<u>Name</u>	<u>Values</u>	<u>Default Value</u>
2.	Task to be performed	TASK	COPY COMP BOTH	BOTH

COPY will do a copy only; COMP will only compare the two tapes, whereas BOTH will do a copy, rewind both units, and then compare the tapes for identical contents.

3.	Input reel no.	IN	xxxxx	-----
----	----------------	----	-------	-------

Where xxxxx is any string of symbols containing at least one letter. Only the last four characters will be used to form the input label. The default value is four underline (____) characters.

4.	Output reel no.	OUT	yyyyy	-----
----	-----------------	-----	-------	-------

Where yyyyy is any string of symbols containing at least one letter. Only the last four characters will be used to form the output label. The default value is four underline (____) characters.

5.	Number of files	EOF	n	all
----	-----------------	-----	---	-----

Where $0 < n \leq 99999$. If the default is used, all files on the input reel are copied (see terminating rules).

6.	Parity	PAR	ODD EVEN BOTH	BOTH
----	--------	-----	---------------------	------

Copying from 7-track to 7-track and specifying a parity parameter value causes the input tape to be read and the output tape to be written with the requested parity.

Since ODD parity is always implied for 9-track units, copying from 7-track to 9-track or from 9-track to 7-track results in operating on the 9-track tape with odd parity and the 7-track tape with the requested parity.

If a value of BOTH is used, the program will switch parity as required for the 7-track tapes only. However, if the 9-track unit is the input, the program will inhibit any parity switching on the 7-track output unit even if the parity parameter value is BOTH.

	<u>Parameter</u>	<u>Name</u>	<u>Values</u>	<u>Default Value</u>
7.	Data converter	CON	ON OFF	OFF
<p>In copying from 7-track to 7-track, the data converter should be ON only if the 7-track input tape had been generated originally with the data converter on. Specifying the data converter parameter value ON results in reading and writing the 7-track input and output tapes with the converter on.</p> <p>Copying from a 7-track unit to a 9-track unit with data converter off (default condition) will result in padding each output tape byte with two additional zero bits. Although the copy and compare operations proceed normally, the resulting output tape is not in a typically useful form. When the data converter is on, odd parity is forced on the 7-track unit. Thus, 7- to 9-track copying would normally be done only for odd parity input (binary tape) and with the converter on.</p> <p>Copying from a 9-track unit to a 7-track unit and specifying CONverter ON results in reading the 9-track tape with ODD parity (converter off and translator off) and writing the 7-track output tape with converter on, odd parity. Data converter OFF is always implied for 9-track units.</p>				
8.	Translator	TRAN	ON OFF	OFF
<p>Copying from 7-track to 7-track and specifying the TRANslator ON parameter, results in reading the input tape and writing the output tape with the translator on.</p> <p>Since TRANslator OFF is always implied for 9-track units, when copying from 7-track to 9-track, from 9-track to 7-track, or from 9-track to 9-track, the TRANslator ON parameter only affects the 7-track tape.</p>				
9.	Messages	MSGS	ON OFF	ON
<p>Messages controlled by this parameter are concerned with the tape records. These messages appear on the console typewriter.</p>				
10.	Retries	TRYS	n	5
<p>Where $0 \leq n \leq 15$ and specifies the number of times to retry reading a record that has been read with error.</p>				

	<u>Parameter</u>	<u>Name</u>	<u>Values</u>	<u>Default Value</u>
11.	Change of buffer size	BUFF	s	2048

Where s specifies a new buffer size in bytes. This buffer size takes effect on the next initiation of the job.

Examples:

1. TPCOPY T181 T280 IN LL6039 OUT LL401A
Copies and compares all files from unit 181 (reel designated 6039) to unit 280 (reel designated 401A) at 800 bpi, converter off, translator off and switching parity as required.
2. TPCOPY T182 T181 TASK COPY EOF 5 DEN 5 PAR EVEN
Copies five files from unit 182 to unit 181 at 556 bpi with even parity and converter off and translator off.
3. TPCOPY T283 T281 CON ON DEN 2
Copies and compares all files from unit 283 (7-track) to unit 281 (9-track) reading from 283 with converter on and odd parity enforced at 200 bpi and writing on 281 with odd parity (converter off and translator off) at 800 bpi.
4. TPCOPY T283 T280 BUFF 3000
Sets TPCOPY program to accept records up to 3000 bytes long on its next initiation (see Changing Buffer Size).

Terminating Rules:

The copy or compare phases terminate at whichever one of the following occurs first:

1. Specified number of files (if EOF parameter is given).
2. Five or more successive tape marks.
3. A tape mark is read in the input tape end-of-reel area.
4. End-of-reel is reached on the output tape.

Console Messages: († implies the task terminates)
(* indicates that message printing is controlled by the MSGS parameters)

1. † FAULTY PARAMETER

An invalid parameter name or value has been specified; check the list above and re-initiate task.

2. † UNIT NOT READY
Ready units and re-initiate task.
3. † OUTPUT UNIT FILE PROTECTED
Unprotect output tape reel and re-initiate task.
4. INPUT IS FROM 9-TRACK UNIT
Warning message; see Data Converter and Nine-Track Unit Operations below for reasonable parameter values.
5. OUTPUT IS ON 9-TRACK UNIT
Warning message; see Data Converter and Nine-Track Unit Operations below for reasonable parameter values.
6. TAPE COPY, VERSION 3 (1-66) INPUT _____ OUTPUT _____
Indicates acceptance of the parameters and beginning of the tape operations.
7. COPY PHASE
Indicates initiation of the copy phase of the operations.
8. * END OF FILE NO. ___, NO. RCS. = ____
One or more consecutive tape marks encountered; the number of records indicated are the count from the last occurrence of this condition or from the starting point; this message only occurs during the copy phase.
9. * PARITY SWITCHED RC. NO. = ____
Program switched parity in attempting to read a record without error (done only if parity parameter value is BOTH).
10. * EXCESSIVE LENGTH RECORD RC. NO. = ____
The record read was too large for the program buffer; the buffer sized portion is accepted for copying or comparison; to process the tape correctly, the task should be re-initiated with an increase in buffer size (see Changing Buffer Size).
11. LAST FILE TRUNCATED
Indicates incomplete file copied due to the output tape being filled before input tape emptied; a terminating tape mark is written on the output tape; terminates the copy phase and, if task parameter value is copy, terminates the task.

12. INPUT TAPE COPIED, INPUT _____ OUTPUT _____, END ON (reason),
TOTAL NO. OF RECORDS = _____, NO. OF CONVERTER CHECKS = _____,
NO. OF BAD RECORDS = _____, NUMBER OF LONG RECORDS = _____,
NUMBER OF TIMES PARITY CHANGED = _____.

Indicates normal completion of the copy phase and, if task parameter value is copy, completion of the task; the possible values of (reason) corresponds to the terminating rules and, are as follows:

- a) SPEC'D NO. FILES
- b) MULT. TAPE MARKS
- c) FULL REEL COPIED
- d) OUTPUT UNIT FULL

The total number of records copied, the total number of converter checks, bad records, and the total number of long records and number of times the parity has changed is given.

13. COMPARE PHASE

Indicates initiation of the comparison phase of the operations.

14. * FAULTY COMPARISON RC. NO. = _____

Indicates the size or contents of the input and output records do not match; this message occurs only during the compare phase; if this same record did not result in the "READ WITH ERROR" message during the copy phase, the comparison should be considered invalid (the tape should be re-compared or possibly re-copied).

15. † TERMINAL COUNTS DO NOT CHECK RC. NO. = _____

Indicates that the number of records and/or number of files processed during the compare phase did not agree with those from the copy phase (occurs under task parameter value both only).

16. † COMPARISON COMPLETE, TOTAL NO. OF RECORDS = _____, NO. OF FAULTY COMPARISONS = _____, NO. OF CONVERTER CHECKS = _____, NO. OF BAD RECORDS = _____, NUMBER OF LONG RECORDS = _____, NUMBER OF TIMES PARITY CHANGED = _____.

Indicates normal completion of the compare phase and the task. The total number of records compared, the total number of faulty comparisons, the total number of converter checks, bad records, long records, and the number of times the parity has changed is given.

17. † INPUT UNIT NOT AVAILABLE

This message is given in two cases:

- 1) if the unit becomes unavailable during rewind; press START on the tape unit to continue with rewind and comparison.
- 2) if the input unit becomes unavailable during the copy or compare process and the resulting tape position is uncertain (e.g., during data transfer); if this occurs in the copy phase, a terminating tape mark is written on the output unit; in this case the task is terminated and should normally be re-initiated from the beginning.

18. † OUTPUT UNIT NOT AVAILABLE

Same as above except no tape mark can be written for case 2.

19. * READ WITH ERROR RC. NO. = ____

Program was unable to read a record without error; program accepts the bad record image for copying or comparison as re-read with the original parity.

20. TAPE Txxx NOT READY

The indicated unit has become unavailable during the copy or compare process without loss of tape position information; press START on the tape unit to allow the task to proceed.

21. † ERROR ON (operation)

Possible value of "operation" here are SENSE, MODE SET, BACKSPACE, and REWIND; indicates possible machine error (CE should be notified); task is terminated and should be re-initiated.

22. * CONVERTER CHECK RC. NO. = ____

A converter check results from reading a 7-track tape with converter on when the number of characters read cannot be stored in an integral number of bytes. For further information refer to IBM 2401, 2402, 2403, 2404, and 2816 Model 1 Principles of Operation Manual, A22-6866.

Additional Console Messages:

Other terminating messages may be generated by the standard tape unit check routines which are used by this program if unresolvable I/O errors occur. Refer to the description of the Tape Unit-Check Routine for the interpretation of these messages.

Changing Buffer Size:

The use of the parameter pair (BUFF n) at task initiation causes the TPCOPY program to require data buffers n bytes long on its next initiation. Thus to copy or compare a tape with records larger than the standard buffer size (now 2048 bytes) requires issuing two task initiation requests, one to change the buffer size and one to execute with the new buffer size. Subsequent task initiations will revert back to the standard buffer size. The value n is not accepted as valid if it exceeds the system's maximum buffer size. However, even if n is not greater than the system's maximum buffer size, two buffers of n bytes may not be available (due to current use by other tasks). In this case, no further initiation of TPCOPY tasks will be possible until two buffers of size n become available. If this situation occurs, the buffers can be changed to a smaller size using the CHANGBUF job.

Changing Density/Parity of 7-Track Tapes:

Copying with data converter off provides one way of copying a 7-track tape onto another 7-track tape at a different parity or density in two steps:

1. 7-track to 9-track at the original parity and density
2. 9-track to 7-track at the new parity and density.

TZDUMP

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

BCD/Binary Tape Dump

Revised 25 November 1966

General Information:

The T7DUMP job dumps a 7-track tape generated on the 1401 or 7094 in BCD or Octal format. This job replaces the function previously performed on the 1401 with the BCD/Binary dump tape program. T7DUMP also dumps in octal format, an even parity 7-track tape.

Options such as mode of tape, density of tape, or number of files to be dumped may be specified. Either BCD, binary, or even parity tapes may be dumped. In the case of a mixed mode tape (BCD and binary) the job determines the mode and chooses the format for printing.

Currently, the maximum record size which can be dumped is 2000 characters. This buffer size for the tape records may be changed prior to execution using the CHANGBUF program.

The system tape unit check routines TPRDUC, PRTUC, and PRTUE are used.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: T7DUMP

Units: PTRp Txxx

where p is the address of a printer and xxx is the address of a 7-track tape. If there is only one printer attached to the system, the printer unit is not specified.

Parameters:

Density:	DEN 2 (for 200 bpi)	Default: DEN 8
	DEN 5 (for 556 bpi)	
	DEN 8 (for 800 bpi)	
No. of Files:	FILE n(where n is > 0)	Default: FILE 1
Mode:	BCD	Default: Mixed mode
	BIN	
	EVEN	

If the tape is dumped in BCD, each 6-bit character on the tape is printed according to the 7094 BCD character code. If the tape is dumped in binary (BIN), or even parity (EVEN), each 6-bit character is printed as two octal digits. In the case of mixed mode tapes, the job will switch modes as required.

Operating Notes:

1. Before each record is printed, the following line is printed:

FILE NO. xxxxxx RECORD NO. yyyyyy ^{BCD}BINARY PRINTED IN OCTAL^{zzzzzz} CHARACTERS
EVEN PARITY PRINTED IN OCTAL

where xxxxxx is the file number, yyyyyy is the record number, and zzzzzz is the number of characters in the record. The format of record is indicated after the record number.

If the record to be printed exceeds the maximum buffer size, the printed record is truncated and the above message is followed by:

--TRUNCATION OF ORIGINAL RECORD SIZE--

2. After each BCD line, a line is printed containing asterisks every ten characters to aid in counting the number of characters in the line.

Examples:

1. T7DUMP T180 DEN 8 FILE 2

Two files from the 800 bpi mixed mode tape on unit 181 are dumped.

2. T7DUMP T180 BIN DEN 5

One file from the 556 bpi tape on unit 180 is dumped in octal format.

3. T7DUMP T181 DEN 2 BCD

One file from the 200 bpi tape on unit 181 is dumped in BCD format.

4. T7DUMP T181 DEN 8 EVEN

One file from the even parity 800 bpi tape on unit 181 is dumped in octal format.

Console Messages: († implies task terminates)

1. † ERROR IN PARAMETERS

If an erroneous parameter is typed, the task will be terminated.

2. END OF FILE
† END OF JOBS

indicates that the task has completed

3. END OF TAPE
† END OF JOBS

indicates that the end of tape mark has been reached on the dump tape.
The task is terminated.

4. PARITY

When a permanent redundancy is encountered, the task prints the following line on the dump listing:

****PARITY ERROR IN READING NEXT RECORD AFTER TEN RETRIES****

If the tape is being printed in mixed mode, the redundant record is printed in both octal and BCD format.

5. FILE MARK WITH PARITY ERROR

If a parity error occurs in reading a file mark on the tape, the program writes the message on the console and on the dump listing. If additional files are to be dumped, the program increments the file count and continues, otherwise it terminates.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Hex/EBCDIC Tape Dump (Re-Entrant)

Revised 25 November 1966

General Information:

The T9DUMP job dumps a 9-track tape or a 7-track tape (recorded with density 800, odd parity, converter on) generated on the 360. The dump can be in hexadecimal or hexadecimal and EBCDIC, and the number of files to be dumped may be specified. The system tape unit check routines TPRDUC, PRTUC, and PRTUE are used.

Currently, the maximum record size which can be dumped is 256 characters. The buffer size for the tape records may be changed prior to execution using the CHANGBUF program.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: T9DUMP

Units: PTRp Txxx

where p is the address of a printer and xxx is the address of a tape unit. If only one printer is attached to the system, the printer unit is not specified.

Parameters:

No. of files: FILE (any number >0) Default: FILE 1

Format: EBCD Default: Hex only

If the parameter EBCD is specified, the dump will be printed in both hexadecimal and EBCDIC. If the parameter EBCD is not specified, the dump will be printed in hexadecimal only.

Operating Notes:

1. Before each record is printed, the following line is printed:

FILE NO. xxxxxx RECORD NO. yyyyyy HEXADECIMAL DUMP zzzzzz CHARACTERS

where xxxxxx is the file number, yyyyyy is the record number, and zzzzzz is the number of characters in the record.

If the record to be printed exceeds the maximum buffer size, the printed record is truncated and the above message is followed by:

--TRUNCATION OF ORIGINAL RECORD SIZE--

Examples:

1. T9DUMP T280 FILE 5 EBCD

Five files from the tape mounted on unit 280 are dumped in both hexadecimal and EBCDIC.

2. T9DUMP T280 FILE 2

Two files are dumped from tape 280 in hexadecimal only.

3. T9DUMP T280

One file is dumped from tape 280 in hexadecimal only.

Console Messages: († indicates task terminates)

1. † ERROR IN PARAMETER

If an erroneous parameter is typed, the task will be terminated.

2. END OF FILE
† END OF JOBS

Indicates that the task has completed.

3. END OF TAPE
† END OF JOBS

Indicates that the end of tape mark has been reached on the dump tape. The task is terminated.

4. PARITY

When a permanent redundancy is encountered, the task prints the following line on the dump listing:

PARITY ERROR IN READING NEXT RECORD AFTER 10 RETRIES

5. FILE MARK WITH PARITY ERROR

If a parity error occurs in reading a file mark on the tape, the program writes the message on the console and on the dump listing. If additional files are to be dumped, the program increments the file count and continues, otherwise it terminates.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Tape Label Program

Revised 5 October 1966

General Information:

LABEL performs the task of writing a label on a 7-track tape (even parity, translator on, density 200, 556, or 800) or on a 9-track tape (odd parity, density 800) on the 360. The label record is a single record followed by a tape mark (EOF). The system tape unit check routine TPWRUC is used.

The job program obtains information from the operator's console to form the volume label sequence number, a 6-character tape label consisting of the 3-character tape bin number followed by the last three digits of the tape reel number, and the tape owner's identification.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: LABEL

Units: Txxx

Where xxx is the address of either a 7-track or 9-track tape.

Parameters:

Tape Reel Number: RELxxx

where xxx is the last three digits of the tape reel number.

Tape Bin Number: BINxxx

where xxx is the tape bin number.

Density: DEN2 (200 bpi) Default: DEN8
 DEN5 (556 bpi)
 DEN8 (800 bpi)

Volume Label Sequence Number:

VOLx Default: VOL1

where x is a volume label sequence number from 1 to 8.

Tape Owner's Identification:

ID Default: LINCOLNLAB

If the ID parameter is typed, the program will type a message on the console typewriter requesting the operator to type a tape owner's

identification of up to ten characters. The system will then type the task number and the job name followed by three dots to indicate that it is the operator's turn to type. A line in which the ID LINCOLNLAB was entered would appear as follows:

00013 LABEL...lincolnlab

Examples:

1. LABEL T280 BINA1B REL649

A label of A1B649 is written on tape 280 with density 800 bpi. The volume label sequence number VOL1 and identification LINCOLNLAB is used.

2. LABEL T280 BINA3J REL345 DEN5

A label of A3J345 is written on tape 280 with density 556 bpi. The volume label sequence number VOL1 and identification LINCOLNLAB is used.

3. LABEL T280 BINB2K REL876 VOL2 ID

A label of B2K876 and a volume label sequence number of VOL2 is written on tape 280 with density of 800 bpi. The tape owner's identification is obtained from the operator.

Console Messages: († indicates task terminates)

1. † ERROR IN PARAMETER

If an erroneous parameter is typed, the task will be terminated.

2. † BIN NUMBER NOT TYPED RESTART JOB

Indicates that the parameter BINxxx has not been typed. The task terminates.

3. † REEL NUMBER NOT TYPED RESTART JOB

Indicates that the parameter RELxxx has not been typed. The task terminates.

4. † END OF TAPE REEL

Indicates that the end of tape mark has been reached on the tape. The task terminates.

5. † PERMANENT WRITE ERROR TAPE PARITY

Indicates that a permanent write redundancy has been encountered in attempting to label the tape. The task terminates.

6. OPERATOR TYPE ID

Indicates that the operator must now supply a 10 character tape owner's ID.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Calcomp Plot Job

27 May 1966

General Information:

CALCOMP performs the task of plotting standard Calcomp output tapes produced on the 7094 or 360. Each picture or reference point on the tape is separated by a header record. Thus, the operator may skip to any specific picture within a file. This job uses the tape unit check routine, TPRDUC.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: CALCOMP

Units: Txxx PLTn

Where xxx is address of 7-track tape to be plotted and n is the number of the plotter to be used.

Parameters:

File: Fn

This parameter, if present, will skip to the file on the tape specified by the integer n.

Picture: Pn

This parameter, if present, will skip to picture or reference point specified by the integer n. The tape will first move to the proper file number.

Example:

1. CALCOMP T281 PLT1

To plot a tape mounted on tape unit 281 on plotter 1.

2. CALCOMP T282 PLT2 P5

To plot a tape mounted on tape unit 282 on plotter 2 beginning at the 5th picture.

3. CALCOMP T180 PLT2 F3 P10

To plot a tape mounted on tape unit 180 on plotter 2 beginning at the 10th picture of the 3rd file.

Console Messages: († indicates task terminates)

1. † PICTURE NOT FOUND
A file mark was reached before the picture specified was found.
2. † INVALID TAPE HAS BEEN MOUNTED. RESTART JOB.
An invalid Calcomp plot tape has been mounted. Check the tape.
3. FILE NO. xxx, PICTURE NO. ppp
This message is printed when a new picture is encountered. The operator may use this as a guide to restarting the job if one of the pictures is ruined by mechanical failure of the plotter. The operator should find the file no. and the picture no. of the last good plot, terminate the current task, rewind the tape, and then restart the job using the proper picture no.
4. † END OF JOB
Occurs when two successive EOF's are read.
5. PERM READ REDUNDANCY
When this message appears, an error mark is written on the plotter to indicate to the programmer that his plot is in error. The error mark is: 'γ'.
6. BAD TAPE, SKIPPING TO NEXT PICTURE. REPOSITION PEN.
This message appears when ten bad records were encountered within one picture. The operator should move the pen, and hit start and stop on the plotter.
7. † EXCESSIVE READ REDUNDANCIES, JOB TERM.
When 50 bad records were encountered, the task will terminate.
8. † ERROR IN PARAMETERS, RECALL.
The operator made a typing error, retry.
9. CALCOMP NOT READY
The plotter associated with this task is not in ready status.

10. END OF FILE. REPOSITION PEN.

The operator should move the pen and paper then hit start and stop on the plotter interface. This will insure that jobs are separated properly.

11. † FAILURE UX.

Inform the system programmer.

12. † UNIT CHECK SENSE X.

Inform the system programmer.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Change Buffer Size Program

6 June 1966

General Information:

CHANGBUF performs the task of changing buffer sizes specified in a job header of a job program. The job program name, buffer numbers, and new buffer sizes are specified as parameters from the console typewriter. Several buffers may be changed at one initiation of the CHANGBUF program. CHANGBUF uses the LLMPs external sub-routine FJBLST, to locate the address of the job program from the job list entry.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: CHANGBUF

Units: No units required

Parameters:

Jobname: jnjnjjnj

The job program name must be specified before all other parameters and may have eight or less characters. If there are more than four characters to a job name, the operator types the first four characters followed by a space, then the remaining characters.

Buffer Number: BFNO n Default: BFNO 1

Where n is the number of the buffer in the job header of the job program. If no buffer number is specified, the program assumes buffer number 1.

Buffer Size: SIZE nnnnn Default: SIZE maxbuf

Where nnnnn is the number of bytes of the new buffer size. If no buffer is typed, CHANGBUF assumes the maximum buffer size allowed by the supervisor.

Operating Notes:

1. The buffer numbers and buffer sizes for a job are specified as keyword and value pairs. The keyword/value pairs of BFNO and SIZE are assumed to be associated but may be specified in any order. If associations are not made explicitly by consecutively specifying both a BFNO and SIZE, the default option is assumed for the keyword/value pair not specified directly.

2. CHANGBUF inserts new buffer sizes into a job header of a job program in core. CHANGBUF does not reset original buffer sizes after execution of the job program; therefore, the job programs should set the buffer sizes to a minimum size after noting the size used for job initiation.

Examples:

CHANGBUF T7DU MP SIZE 10000 BFNO 1

The size of the first buffer in the job program T7DUMP is changed to 10,000 bytes.

CHANGBUF T7DU MP

The size of the first buffer of the job program T7DUMP is set to the size of the largest buffer in the system.

CHANGBUF PRIN T SIZE 2000 SIZE 5000 BFNO 2

The size of the first buffer is changed to 2000 bytes, and the size of the second buffer is changed to 5000 bytes in the job program PRINT.

Console Messages: († indicates task terminates)

1. † ERROR IN PARAMETER
If an erroneous parameter is typed, the task terminates.
2. † BUFFER SIZE EXCEEDS MAXIMUM BUFFER
The buffer size specified exceeds the maximum buffer allotted by the supervisor. The task terminates.
3. † ZERO BFNO NUMBER TYPED OR NUMBER EXCEEDS SIX
The buffer number typed is zero or exceeds the maximum number of buffers allowed by LLMPs. The task terminates.
4. † JOB NOT FOUND IN JOB LIST
The job program is not in the current system. The task terminates.
5. † BUFFER TO BE CHANGED EXCEEDS TOTAL NUMBER OF BUFFERS IN JOB HEADER
The buffer number typed exceeds the total number of buffers specified in the job header of the indicated job program. The task terminates.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Disk Dump Job

17 June 1966

General Information:

The DISKDUMP job displays the contents of a 2311 disk on a printer. Records on the disk may be dumped in both hexadecimal and EBCD or, alternatively, the standard Volume Table of Contents (VTOC) may be displayed.

Each record which is displayed is preceded by an identifying label which gives the address of the record (CYL - TRK - REC) and also the key length and the data length. The key length and data length are obtained from the count area which is the first 11 bytes of a record. The key length is specified in the 7th byte of the count area and the data length is specified in the 8th and 9th bytes of the count area. The count area is followed immediately by the key and data areas which are printed out in both EBCDIC and hexadecimal.

To display the Volume Table of Contents, the address of the VTOC is obtained from the volume label which is assumed to be on cylinder 0, track 0, record 3 of the disk. The volume label has "VOL1" in bytes 1 thru 4 of the label record and the volume ID in bytes 5 thru 10. The address of the VTOC is obtained from bytes 12 thru 16 of the label record and is in the form CCTTR where CC is the cylinder number (0-201), TT is the track number (0-9), and R is the record number (1-20).

The information displayed from the VTOC is obtained from Format 1 records (indicated by an F1 in the 45th byte of the record) and the extent of the VTOC is obtained from the Format 4 record (indicated by an F4 in the 45th byte of the record). This job program assumes that the Format 4 record is the first record of the VTOC. The address of the last Format 1 record is obtained from bytes 46 thru 50 of the Format 4 record in the form CCTTR. Format 1 records are read until the last one, indicated by the address specified in the Format 4 record, is reached. The information displayed from Format 1 records consists of a data set name (bytes 1-44), the starting address of the data set (bytes 108-111), and the ending address of the data set (bytes 112-115). The addresses are specified in the form CCTT.

For more information on the 2311 disk refer to IBM System/360 Component Descriptions - Form A26-5988 and Introduction to Control Program Logic - Form Z28-6605.

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: DISKDUMP

Units: PTRr Dxxx

Where r is on the address of a 1403 printer and xxx is the address of the 2311 disk to be dumped. If there is only one printer attached to the system, the printer ID does not have to be specified.

Parameters:

Beginning Address:

cyl trk rec Default: VTOC

The address of the first record to be dumped is specified by 3 decimal parameter indicating the cylinder, track, and record numbers. Either all of these first 3 parameters must be specified or none of them. If no parameters are specified, the Volume Table of Contents is displayed.

Ending Address:

maxcyl maxtrk Default: entire disk

Where MAXCYL and MAXTRK indicate the cylinder and track number at which the program is to stop dumping. The task will terminate after printing the last record on the track preceding the one identified by MAXTRK. If specified, these two parameters must follow the three parameters indicating the beginning address. If they are not specified, the disk will be dumped beginning at the address specified by the first three parameters through to the end of the disk.

Examples:

1. DISKDUMP D390

The VTOC on disk 390 will be printed giving data set names, beginning addresses, and ending addresses.

2. DISKDUMP D390 2 3 4

The entire disk 390 will be dumped beginning with cylinder 2, track 3, and record 4.

3. DISKDUMP D390 2 3 4 2 4

The records on cylinder 2, track 3 of disk 390 will be dumped beginning with record 4 (the 5th record).

Console Messages: († indicates job terminates)

1. DEVICE NOT READY. READY DISK Dxxx
 When the disk is made ready the job will continue.
2. † END OF TABLE OF CONTENTS
 After the VTOC has been displayed, this message will be printed
 on the console typewriter and the job terminated.
3. † NO RECORD FOUND. JOB TERMINATED.
 An invalid record address has been specified.
4. † THIS IS A DEFECTIVE TRACK. UNABLE TO READ THIS TRACK.
 An unrecoverable read error has occurred.
5. † INVALID SEEK ADDRESS. JOB TERMINATED.
 A possible program error has occurred.

The following console messages may indicate hardware failures.

6. † BUS OUT PARITY CHECK. JOB TERMINATED.
7. † COMMAND REJECT. JOB TERMINATED.
8. † DATA CHECK. JOB TERMINATED.
9. † MACHINE ERROR ON Dxxx.

HINPILOAD-
UCSLOAD

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

HNPLOAD and UCSLOAD Program

5 October 1966

General Information:

The HNPLOAD and UCSLOAD program loads an arrangement of 240 codes selected from the Extended Binary Coded Decimal Interchange Code into the read/write storage units of the 2821 control unit for a 1403 printer. The program requires a 1403 printer and 2821 control unit with the universal character set feature. Any user-designed chain or train arrangement may be loaded in addition to the following IBM arrangements: PCAN, PCHN, AN, HN, PN, QN, RN, SN, TN, XN, and YN.

The HNPLOAD program can be used to load a 1403 printer control unit buffer with the codes for either an HP or PN train. The codes are stored within the job program and are transmitted to the control unit.

The UCSLOAD program requires that a NAME control card and four IMAGE control cards be provided as described below. The NAME card identifies the character set of the printer chain (train) and must precede the IMAGE cards which contain the desired 240 character to be loaded into the control unit buffer.

If a user-designed arrangement is to be loaded, the user must adhere to the restriction that graphics may be arranged in any sequence provided no character repeats within 16 positions. Further information relating to the designing of an arrangement may be obtained from IBM manuals C24-3396, Universal Character Set Utility Program, and A24-3312-2, IBM 2821 Control Unit.

The system tape unit check routines CDRDUC, CDSTUC, PRTUE, and PRTUC are used. Both the USCLOAD program and the HNPLOAD program are re-entrant.

Task Request Formats:

Jobname: HNPLOAD

Units: PTRp

Where p is the address of a 1403 printer with the UCS feature.

Jobname: UCSLOAD

Units: RDRr PTRp

Where r is the address of a 2540 card reader and p is the address of a printer. If there is only one reader or printer attached to the system, the reader or printer unit is not specified.

Parameters:

1. Folding: FOLD Default: No Folding

Folding of the entire third quadrant of the EBCDIC code into the fourth quadrant causes incoming upper and lower case code data to be printed out using only an upper case arrangement. If the power is turned off in the 2821 control unit, the folding feature is lost, but the contents in the storage unit are retained.

2. Train: HN Default: HN
PN

This parameter specifies the train IMAGE to be used. The codes for these arrangements are stored within the job program. This parameter should only be used with the HNPLOAD program; it causes a parameter error if used with the UCSLOAD program.

UCS Program Control Cards (For UCSLOAD program only):

NAME Card:

Function: Identifies character set used

Usage: One per character set

Format: Fixed, as follows:

<u>Cols.</u>	<u>Punch</u>	<u>Purpose</u>
1 - 2	//	Identifies control card
3	blank	Separator
4 - 7	NAME	Operation
8 - 10	blank	Not used
11-80	Application Name	Identifies character set

IMAGE Cards:

Function: Provides characters to be loaded
Usage: Four cards per character set
Format: Fixed, as follows:

<u>Cols.</u>	<u>Punch</u>	<u>Purpose</u>
1 - 2	//	Identifies control card
3	blank	Separator
4 - 5	CI	Operation
6	blank	Separator
7 - 10	character coded identification	Character set code identification
11	blank	Separator
12	S	Sequence number follows
13	[1 for card 1 2 for card 2 3 for card 3 4 for card 4	Sequence number of individual CI cards in ascending sequence
14	blank	Not used
15	blank for CI cards 1, 2, and 3	Card 4 contains the number of IMAGE lines to be printed (See Table III. B. 14-2)
16-20	blank	Not used
21-80	characters to be loaded into the UCS buffer	Each CI card contains 60 characters

Operating Notes:

When loading is complete, the program prints an identifying message, if an IBM arrangement is used, or the character sequence of the printer chain (train), if a user-designated arrangement is used, followed by the chain (train) IMAGE, and then the contents of the NAME card - provided that the NAME card can be printed with the set of characters on the printer chain (train).

If an IBM arrangement is used and the specified chain (train) was mounted, an identifying message as indicated in Table III. B. 14-1 will be printed. The operator should verify that the message printed is correct.

If a user-designated arrangement is used, the operator should verify that the character sequence printed corresponds to the desired chain (train) arrangement.

The number of printed lines in the chain (train) image is specified in column 15 of the fourth IMAGE control card and is indicated in Table II for the IBM arrangements. The number of lines corresponds to the repeat factor of the chain (train).

In Tables III. B. 14-1 and III. B. 14-2, the XN arrangement is for a chain cartridge only, and the YN arrangement is for a train cartridge only.

An example of the complete printout for the IBM HN arrangement is shown below:

HN ARRANGEMENT TYPE IF EQUAL SIGN =

1234567890=' / STUVWXYZ#, (JKLMNOPQR-\$*ABCDEFGHI+.)

1234567890=' / STUVWXYZ#, (JKLMNOPQR-\$*ABCDEFGHI+.)

1234567890=' / STUVWXYZ#, (JKLMNOPQR-\$*ABCDEFGHI+.)

1234567890=' / STUVWXYZ#, (JKLMNOPQR-\$*ABCDEFGHI+.)

1234567890=' / STUVWXYZ#, (JKLMNOPQR-\$*ABCDEFGHI+.)

LL HN PRINT CHAIN

<u>Cartridge</u>	<u>Identifying Message</u>
PCAN	PCAN ARRANGEMENT TYPE IF POUND SIGN #
PCHN	PCHN ARRANGEMENT TYPE IF EQUAL SIGN =
AN	AN ARRANGEMENT TYPE IF POUND SIGN #
HN	HN ARRANGEMENT TYPE IF EQUAL SIGN =
PN	PN ARRANGEMENT TYPE IF POUND SIGN #
QN	QN ARRANGEMENT TYPE IF POUND SIGN #
RN	RN ARRANGEMENT TYPE IF POUND SIGN #
SN	SN ARRANGEMENT TYPE IF COMMA SIGN ,
TN	TN ARRANGEMENT TYPE IF POUND SIGN #
XN	XN ARRANGEMENT TYPE IF COMMA SIGN ,
YN	YN ARRANGEMENT TYPE IF POUND SIGN #

Table III. B. 14-1

System/360 Cartridges
(Punched in CI cards 1, 2, 3 and 4)

Number of printed lines in the
chain (train) Image
(Punched in CI card 4 only)

<u>Cols.</u> 7 8 9 10	<u>Col.</u> 15
P C A N	4
P C H N	4
A N	5
H N	5
P N	4
Q N	5
R N	5
S N	A (3 lines will be printed)
T N	2
X N	6
Y N	B (6 lines will be printed)

Table III. B. 14-2

Examples:

1. HNPLOAD PTRF

The IBM HN cartridge train arrangement is loaded into the 2821 control unit buffer for printer F

2. HNPLOAD PTRF PN FOLD

The IBM PN cartridge train arrangement is loaded into the 2821 control unit buffer for printer F with the folding option.

3. UCSLOAD PTRE

The 240-character set is loaded into the 2821 control unit for printer E.

4. UCSLOAD RDRr PTRp FOLD

The 240-character set is loaded into the 2821 control unit with folding capability.

Console Messages: († indicates task terminates)

1. † ERROR IN PARAMETER
If an erroneous parameter is typed, the task terminates.
2. † IMPROPER CONTROL CARD
If a control card doesn't have a slash in columns 1 and 2 and a blank in column 3, the task terminates.
3. † UNIDENTIFIABLE CARD READ
Indicates that the program was unable to identify the type of control card read. The task terminates.
4. † CARD SEQUENCE ERROR
Indicates that the NAME control card was not the first card read or that the IMAGE control cards are not in ascending order. The task terminates.
5. † CI CARDS NOT FROM SAME DECK
The chain image control cards do not have the same 4-character identification code. The task terminates.
6. † CARD PUNCHED INCORRECTLY
Columns 14-20 of the chain image cards must be blank. If not, the above message is given and the task is terminated.
7. † CHARACTER REPEATS WITHIN 16 POSITIONS
Graphics may be arranged in any sequence on the print chain/train provided that no character repeats within 16 positions. If a character repeats within 16 positions, the task terminates.
8. CHECK VALIDITY OF CHAIN/TRAIN
When loading is complete the above message is typed to the operator.

CONVERT-
REPRO

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

CONVERT/REPRO Job

Revised 3 November 1966

General Information:

CONVERT accepts cards punched with the codes produced by an IBM Model 026 card punch (the pre-360 BCD code) and produces new cards, on a one-for-one basis, that are punched in accordance with the EBCDIC code used by S/360. This job is useful in converting cards punched on the Model 026 Card Punch to cards that would have been punched on the (newer) Model 029 Card Punch if keys with the same graphics had been used.

REPRO produces new cards on a one-for-one basis. The reproduced cards are identical to the input cards regardless of the codes used. Input codes are not checked for validity; therefore, the job may be used to duplicate any deck.

Both programs use the CDRDUC, CDSTUC and CDPNUC routines for card unit checks. Both programs are re-entrant and may be used simultaneously without conflict provided the necessary readers and punches are available.

Task Request Summary:

CONVERT RDRr PCHp

REPRO RDRr PCHp

Sense Switch Bits Used: 2 (EOF)

Task Request Format: JOBNAME UNITS PARAMETERS

Jobname: CONVERT or REPRO

Units: RDRr PCHp

where r is the address of a 2540 card reader and p is the address of a 2540 card punch. If there is only one card reader or one card punch attached to the system, the reader or punch unit need not be specified.

Parameters: None

Examples:

CONVERT

Any cards placed into the reader hopper will be reproduced with converted codes by the punch.

REPRO

Any cards placed into the reader hopper will be reproduced without change.

Sense Switch Usage:

Bit 2: If bit 2 of the sense switch is set, the task will terminate after the last card has been read and reproduced, provided that the End-of-File light is set. If this switch bit is not set, or the End-of-File light is not set, the task will not terminate.

Console Messages:

1. I/O ERROR

A unit exception occurred on the card punch. A job dump occurs and the task is terminated.

Operating Instructions:

The task feeds one blank card from the punch before punching, and feeds one final blank card from the punch after having reproduced the last card read with the end-of-file light on. It is therefore never necessary to perform a non-process run-out of the punch. After having converted (or reproduced) one deck, another deck will be processed whenever the reader is made ready again. The task can only be terminated by setting switch 2 with the End-of-File light set, or by the TERM task.

Conversion Routine: (Used by CONVERT only)

Any card with an acceptable EBCDIC punched card code will be read and converted. 251 of the possible codes will be reproduced unchanged. The five codes in the table below are converted as shown. Cards requiring the column binary feature are not acceptable to the CONVERT program.

Model 026 Code			converts to	Model 029 Code		
BCD Graphic	Punched Code	Hex Code		EBCDIC Graphic	Punched Code	Hex Code
)	12-8-4	4C)	11-8-5	5D
+	12	5O		+	12-8-6	4E
(0-8-4	6C		(12-8-5	4D
=	8-3	7B		=	8-6	7E
'	8-4	7C		'	8-5	7D

Table III. B. 15-1. 026 to 029 Conversion Table.

Program Logic:

The following flow charts describe the logical processing of the job.

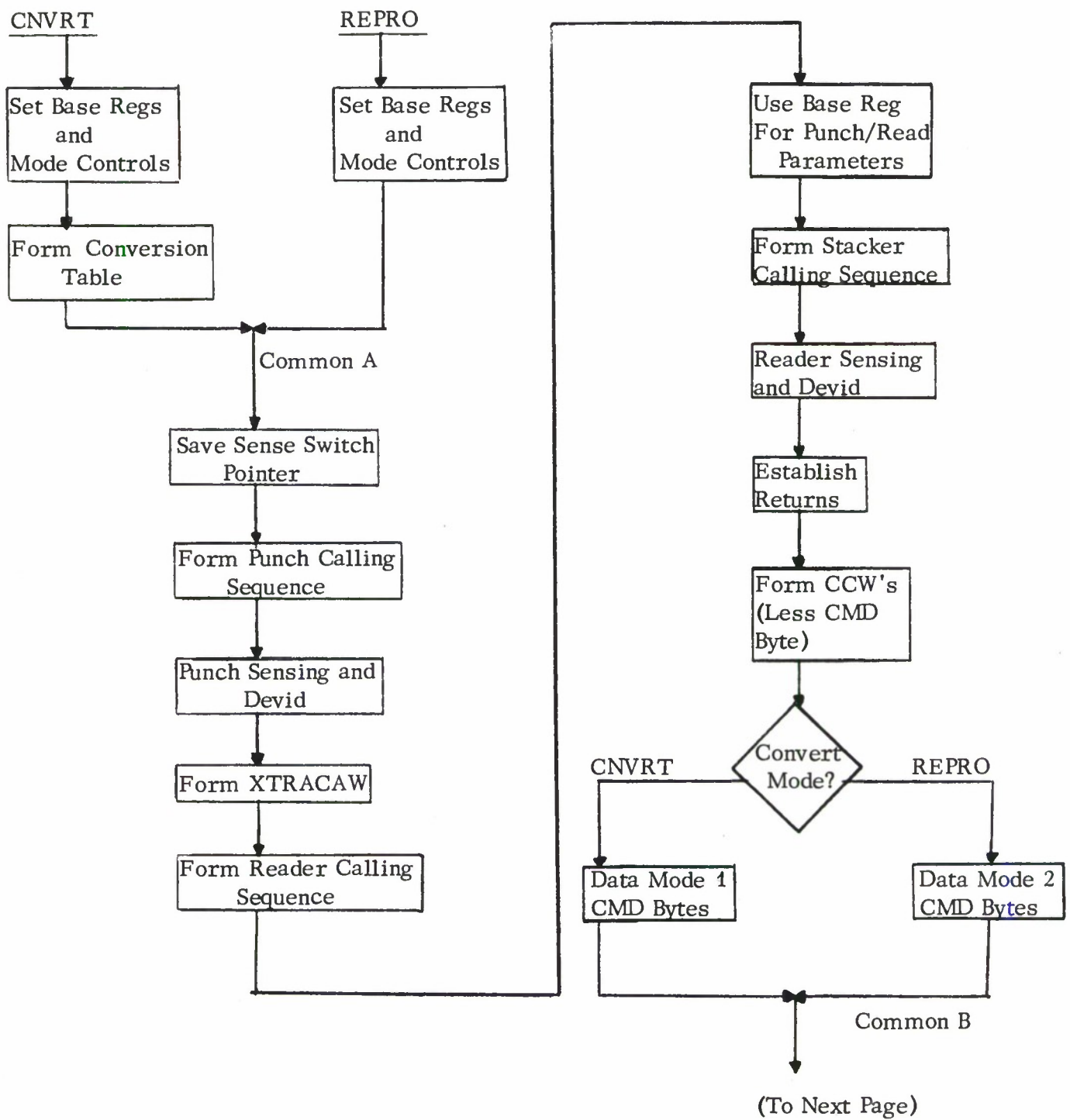


Fig. III. B. 15-1. CONVERT/REPRO Flow Diagram.

(From Previous Page)

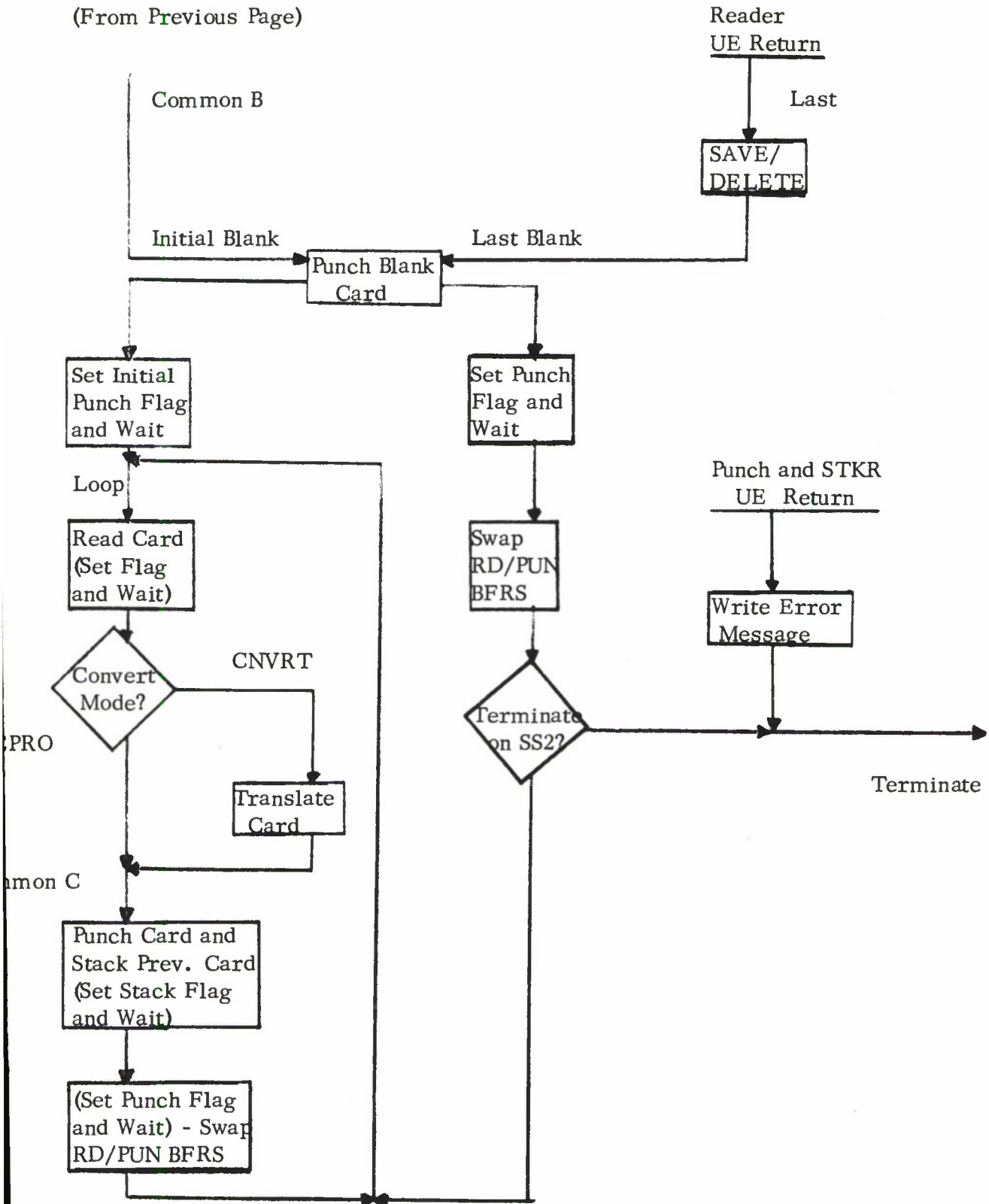


Fig. III. B. 15-1. CONVERT/REPRO Flow Diagram (Cont.)

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

Paper Tape Punch Job

21 November 1966

General Information:

PUNCHTAP reads card images in EBCDIC code from a 9-track tape (or from a 7-track tape with density 800, odd parity, converter on) and punches them out on paper tape in FIELDATA code. The paper tape produced can be used as input to the Univac 1219.

The EBCDIC card images are handled by PUNCHTAP as follows. Card columns 11, 21 and 39 are considered delimiters of fields 1, 2 and 3 respectively. If a delimiter is blank it is replaced by a tab and the trailing blanks of the preceding field are truncated. If a delimiter is not blank, the preceding field is left as is. Trailing blanks are also suppressed in the "comment" field (columns 40-80). The hex equivalent for carriage return-line feed (hex FA, FB) are then inserted at the end of the compressed card image. Each character in that image is converted to FIELDATA code as shown in the translation table and punched out on paper tape. Note that the program converts the EBCDIC character @ (8-4 punch) to the FIELDATA stop code and punches a stop code followed by 3 inches of blank tape.

The program uses the TPRDUC routine for tape read unit checks.

Task Request Summary:

PUNCHTAP Txxx PLTh

Task Request Format:

Jobname: PUNCHTAP

Units: Txxx PLTh

where xxx is the address of a 9-track tape (or 7-track tape recorded with density 800, odd parity, converter on) containing the EBCDIC card images and PLTh is the address of the paper tape punch. Currently, PLT2 must be used for the paper tape punch.

Parameters: None

Operating Note:

The power switch on the paper tape interface box must be on and the rotary switch on the Calcomp interface box must be in the paper tape position.

Example:

PUNCHTAP T180 PLT2

The EBCDIC card images on the tape mounted on tape unit 180 will be punched on paper tape in FIELDATA code.

Console Message:

PAPER TAPE PUNCH ERROR aaa. STATUS = b SENSE = cc

Where aaa is either UEX for unit exception or UCK for unit check, b is a number indicating the status of the sense command, i.e., sense byte flag from sense operation (see SVC19-SNSADR), and cc are two hexadecimal numbers representing the sense byte.

FIELDATA Translation Table:

<u>EBCDIC Code</u>		<u>Graphic</u>	<u>FIELDATA Code</u>	
<u>029 Card Code</u>	<u>Hex</u>		<u>Octal</u>	<u>FIELDATA Graphic</u> (where different)
None	40	space	05	
12-8-2	4A	0	01	() Upper case
12-8-3	4B	.	75	
12-8-4	4C	<	43	
12-8-5	4D	(51	
12-8-6	4E	+	42	
12-8-7	4F		77	(↑)
12	50	&	42	(+)
11-8-2	5A	!	55	
11-8-3	5B	\$	47	
11-8-4	5C	*	50	
11-8-5	5D)	40	
11-8-6	5E	;	73	
11-8-7	5F	⌵	76	(□) tab
11	60	-	41	
0-1	61	/	74	
0-8-3	6B	,	56	
0-8-4	6C	%	02	(%) Lower case
0-8-5	6D	—	46	
0-8-6	6E	>	45	
0-8-7	6F	?	54	
8-2	7A	:	53	
8-3	7B	#	00	(⋈) space
8-4	7C	@	57	(Ⓢ) stop code
8-5	7D	'	72	
8-6	7E	=	44	
8-7	7F	"	52	
12-1 thru 12-9	C1 - C9	A thru I	06 thru 16	
11-1 thru 11-9	D1 - D9	J thru R	17 thru 27	
0-2 thru 0-9	E2 - E9	S thru Z	30 thru 37	
0 thru 9	F0 - F9	0 thru 9	60 thru 71	
end of card	FA FB	None	04 followed by 03	

NOTE: Any EBCDIC configuration not on this table will be converted to FIELDATA OCTAL 00 which is not punched but merely advances the paper tape 1 character position.

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instruction

LESTER Job

14 November 1966

Job Summary:

The LES Telemetry Program (LESTER) receives LES telemetry data via one of the LES telephone lines from the tracking station; formats the input data into printable records; and outputs these records onto either a 7 or 9-track tape.

General Information:

The LESTER job reads 960 byte input records from any one of the LES telephone lines. Each input record is converted to EBCDIC code, headed with a 4-byte identifier, formatted for printing, and written onto an output tape as thirty 133-byte records. The tape may later be listed using the LLMPs PRINT job as shown in the sample listing.

In its present form, LESTER will lose the first character of every 960 character input record. This somewhat annoying peculiarity is caused by a supervisor/hardware incompatibility and cannot now be easily fixed. To make this character loss more palatable, the program will synchronize its reading so that after the first 3 records, the lost character of every successive record will be the frame-synchronization character. Also, if synchronization is lost, the routine resynchronizes itself within the next three successive records. This synchronization ability, combined with a large input record size (currently set at 960 bytes), will lessen the effect of the lost characters.

Task Request Summary:

LESTER	Xnnn	Txxx	[Fyyyy (FE349)]
--------	------	------	----------------------

Switch Bit 0: Terminate immediately.

Task Request Format:

Jobname: LESTER

Units: Xnnn Txxx

where nnn is the last 3 digits of the LES extension and xxx is a tape unit address.

The valid LESTER input telephone lines are Lincoln Laboratory extensions 7702, 7704, and 7706. Any one of these lines may be used for transmitting the LES data to the computer. If any of the other computer input lines are used, LESTER will immediately terminate with an intervention required message typed on the console typewriter.

Parameter: Fyyyy

Default: FE349

where yyyy is the hexadecimal representation of one of the pairs of the frame-synchronization characters.

Job Termination:

The LESTER job will terminate with an error message if no frame-synchronization characters are found in the first input record. A normal termination is assumed to have occurred when either the sending data set signals intervention required (goes out of data mode) or when switch bit 0 is set. At all terminations the LES line will be disabled, file marks (EOF) written on the output tape, and the tape rewound and unloaded.

Example:

LESTER X702 T180 F1CBC

The LESTER job is initiated from line 7702 and will write the telemetry data on tape unit 180 using 1CBC as the frame synchronization character.

Console Message:

1. NO SYNC. CHARS. PLEASE NOTIFY LESS STATION.

When the synchronization character cannot be found in the first input record, the message is printed on the console typewriter and the job continues processing ignoring synchronization characters.

Telemetry Input:

Because of its synchronization property, LESTER is extremely dependent upon the format or the input data. The telemetry input is assumed to have the following form:

- i) 2 frame-synchronization characters
- ii) 6 sequences of data with each sequence consisting of 4 data characters followed by a parity character.
- iii) Complement of the previous frame-synchronization characters followed by the above data sequences.

The input sequence then repeats starting with the original frame-synchronization characters. (See Fig. III. B. 17-1).

Values of either pair of frame-synchronization characters are input from the console typewriter when the job is started. The other format properties of the input stream, (i.e., length of sequences etc.), are fixed and LESTER will have to be modified if these are to change.

```
ffddddpddddpddddpddddpddddpddddpf'f'd....pffd....pf'f'd...
```

Telemetry Input Stream

ff = frame synchronization characters f'f' = complement of ff

p = parity character d = data character

All characters consist of 8 bits = 2 hexadecimal digits.

Fig. III. B. 17-1. Assumed Format for Telemetry Input.

LESTER Output Tape:

The LESTER output tape is written in fixed length 133-byte records. Each record is written in EBCDIC code where the initial character of each record is a FORTRAN type carriage-control character. The general format of the tape is a column-heading record followed by 60 data records with the sequence repeating with the column-heading record. The 60 data records represent two 960-byte input records. Each input record is written as 30 output records. The first output record for an input record contains a 4-byte input record identifier followed by the frame-synch. character, the data, and the parity sequence up to the second frame-synch. characters. The lost-character is the initial output character of this sequence and it is always written as **. The subsequent 29 output records repeat the frame-synch. to frame-synch. sequence but eliminate the input record identifier and the lost character marker. The last record on the tape consists of 3 successive tape marks. A columnar description of one sequence of output on the tape is shown below.

During the three input-record synchronization sequences, a slightly different format is used. The first two input records during this sequence are 96 bytes in length. Each of these records is written out as an identifier record followed by 9 data records. Note that only the first 3 records in this output will contain telemetry data. All of the remaining records will be zeroed. Also the lost-data byte of these records may be telemetry data instead of a synch. character.

The third input record will be between 96 and 128 bytes in length. This is written out as an identifier record followed by 29 data records. The lost character will be a frame-synch. character and all non-telemetry output records will be zeroed.

All subsequent input records are 960 bytes in length and are written out in the standard format.

Format of the Column Heading Record :

<u>Item</u>	<u>Columns</u>	
Carriage Control	1	A skip to new page followed by a print and skip 1 line (X'F1')
Heading Information	2-133	Column headings for the output. ID = identifier for input record FS = frame-synch. column D = data column P = parity character column

Format of Initial Output Data Record for an Input Record :

<u>Item</u>	<u>Columns</u>	
Carriage Control	1	A print and skip 1 line character (x'40')
Input Record Identifier	4-11	Columns 4-5 contain the sense byte associated with the input record; columns 6-7 are zero and columns 8-11 contain the sequential input record number for this record.
Lost Character	14-15	This character was lost in the 2702. It is designated by ** in columns 14 and 15.
Frame-Synch. Character	16-17	Starting with input record number 4, these columns should contain the second character of one of the pairs of the frame-synch. characters. On the following output record this character should be complemented.
Data Characters	20-27 34-41 48-55 62-69 76-83 90-97	Data characters written as 2 hexadecimal digits per data character.
Parity	30-31 44-45 58-59 72-73 86-87	Parity character for the preceeding 4 data characters

All non-specified columns are written as EBCDIC blanks (x'40').

Format of Remaining Output Records for an Input Record :

<u>Item</u>	<u>Columns</u>	
Carriage Control	1	Print and skip character (x'40')
Frame-synch.	14-17	Both frame-synch. characters will appear on this and the following 28 output records. These characters will be complemented on each record. If synchronization has been lost, these characters will not have the expected pattern. Synchronization may again be attained on the third input record following this record.
Data Characters	Same as initial record	
Parity Characters	Same as initial record	

Again, all non-specified characters are written as blanks.

Example Listing of a LESTER Output Tape*

[illegible]

***Note:** This listing was obtained when there were telemetry and tape unit hardware problems. As a result, the values of the frame-sync. and data characters may be incorrect. The format, however, is correct.

E389	94939291	80	900F8E80	1E	8C88A89	78	88878685	67	84838281	C3	807F7E70	F6
1CA6	7C787A77	FF	78777675	EC	74737271	4C	706F6E6D	DF	6C686A69	B9	68676665	A6
E301	64036201	0A	605F5E5D	A0	5C581A59	72	58575655	60	74535251	C1	504F4E4D	72
1CB6	4C484A49	34	42FF4746	45	2B444342	41	07803F3E	3D	5F3C383A	79	C4383736	35
DBE3	49343332	31	77302F2E	2D	E42C232A	29	92282726	25	90242322	21	31201F1E	1D
981C	861C161A	19	81181716	15	58141312	11	FA10GF0E	0D	490C080A	09	0F08C706	05
1CE3	87040302	01	6C00FFFE	FD	01FCFBFA	F5	88F8F7F6	F5	97F4F3F2	F1	38F0EFEE	CD
A81C	86EC8E8A	E9	CE8E7E6	E5	D1E4E3E2	E1	70E0DFDE	9D	D79C0BDA	09	0508D7D6	D5
1AE3	01D4D3D2	D1	86D0CFCE	CD	25CC8CA	C9	43C8C7C6	C5	5CC4C3C2	C1	F0C08F8E	8D
281C	858C868A	B9	8388786	B5	AC848382	31	008CAFAE	AD	93ACA8AA	29	F548A706	A5
C2E3	49A4A3A2	A1	46A09F9E	9D	EC9C9B9A	99	3E989716	95	41949392	91	8D900F8E	8D
1E1C	068C088A	89	76888786	85	67848382	01	C8807FFE	7D	F67C7B7A	79	FF787776	F5
E0E3	49747372	71	0C7C6F6E	6D	DF6C6B6A	69	B9686766	65	A6646362	61	0A605F5E	59
A01C	865C865A	59	72585716	E5	6D545392	51	C1504F4E	4D	524C4B4A	49	34484746	45
28E3	49444342	41	07403F3E	3D	5F3C3B3A	39	80383736	35	83343332	31	77302F2E	2D
F41C	362C282A	29	82282726	25	5D242322	21	31201F1E	1D	981C181A	19	49181716	15
56E3	49141312	11	FA10GF0E	0D	695CGB0A	09	0F0807C6	05	10040302	01	8C00FFFE	FD
8118	86FCFBFA	F9	88F8F7F6	F5	97F4F3F2	F1	38F0EFEE	ED	A8ECEBEA	E9	CE8E7E6	E5
01E3	49C4C3C2	E1	70E0DFDE	DD	L7LC0BDA	D9	0598D706	D5	1AD4D3D2	D1	86D0CFCE	CD
251C	86CC86CA	C9	43C8C7C6	85	5CC4C3C2	C1	F0C08F8E	8D	288C8B8A	B9	8388B786	B5
ACE3	09340382	71	0C30AFAE	8D	03ACA8AA	A9	F54807A6	A5	EAA4A3A2	A1	46A09F9E	9D
EC1C	869C869A	59	3E989796	95	21949392	51	0D968F8E	8D	1E8C8B8A	89	78088706	85

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

LOAD Job (Re-entrant)

1 April 1967

Job Summary:

The LOAD program relocates and loads assembled LLMPS programs and program joblists into core.

General Information:

The LOAD program relocates and loads into storage assembled LLMPS job programs and joblist programs. LLMPS job programs and joblist programs may be assembled using IBM BPS or OS/360. The LOAD program allows corrections to be made to assembled programs at load time. It also establishes linkages between separately assembled programs that are resident in core as part of the LLMPS system and the assembled program being loaded. The LOAD program re-evaluates and adjusts address constants used by the assembled program during the loading procedure. The assembled job program must contain only a single control section. Multiple jobs may be contained within the control section. No subroutines, other than those preloaded, may be used by the assembled job program.

Processing of Object Decks:

The LOAD program recognizes as input five types of cards. Four of these are generated by the assembler:

1. EXTERNAL SYMBOL DICTIONARY (ESD)

The ESD card contains the external symbols defined or referred to by the program. Each entry on an ESD card may be classified as: a program segment name, an ESD TYPE 0; an entry point in the program segment, an ESD TYPE 1; and an external name referred to by the program, ESD TYPE 2.

2. TEXT CARD (TXT)

The TXT card contains the instructions and data of the program.

3. RELOCATION LIST DICTIONARY (RLD)

The RLD card contains information about address constants in the program.

4. END CARD (END)

The END card signified termination of the program.

The fifth card, the REPLACE (REP) card is supplied by the programmer to modify the instructions and data of the assembled program at load time. Replace cards are printed on the console typewriter during the loading process. For a detailed description of these cards refer to IBM System/360 Basic Programming Support Basic Utilities, C28-6505.

Storage Map:

The LOAD program lists the program segment and entry point names with their current storage addresses on the console typewriter during the loading process.

Table of External References:

The assembled LLMPs program may refer externally to the following program segments resident in core as part of LLMPs system.

JOBTL	Pointer to start of job tables
MAXADD	Maximum core address
BUFMAL	Maximum buffer size
FNDJTL	Find job table location routine
FBJTL	Free buffers, job table, and joblist routine
FJBLST	Find job list routine
JOBLST	Pointer to start of joblists in core
FIDCQ	Free I/O devices and CPU queue routine
BTOHEX	Binary to hex conversion routine
EBTOXT	EBCDIC to 2741 code conversion routine
XTTOEB	2741 code to EBCDIC conversion routine
HEXBIN	Table for converting from EBCDIC hex characters to binary values
TRANSX	Translates n bytes according to table
CDSTUC	Feed and select stacker UC routine
CDRDUC	Card reader UC routine
CDPNUC	Card punch UC routine
CDPNUE	Card punch UE routine
TPWRUC	Tape write UC routine
TPRDUC	Tape read UC routine
PRTUC	Printer UC routine
PRTUE	Printer UE routine
GOCK	Unit check routine for LESTER
TOBIN	Translates and packs n unpacked characters into (N + 1)/2 bytes according to any table-entry point
HXTOBIN	Hex to binary conversion routine

GOEX	UE routine for LESTER
WTM	Write tape mark routine
ERG	Erase record gap routine
FSR	Forward space tape n records routine
BSR	Backspace tape n records routine
BSF	Backspace tape n files routine
FSF	Forward space tape n files routine
REW	Rewind tape to load point routine
RUN	Rewind and unload tape routine
GETPRM	Locates a parameter in a message for 2741/2702 routine.
XEDT	Performs desired erasures of a message to 2741/2702 routine
XEX	Routine to handle unit exceptions for all commands to 2741/2702
XCK	Routine to handle unit checks for all commands to 2741/2702
XSIO	Constructs I/O commands for 2741/2702 operations routine
RDIT	Reads a message from 2741 routine
TRDIT	Reads a message from 2741 and translates it to EBCDIC routine
SNDIT	Sends a message to the 2741 routine
TSNDIT	Translates (EBCDIC to 2741) and sends a message to 2741 routine

Any other external references made by the assembled program will cause the error message, 'EXTERNAL REFERENCE NOT IN TABLE', and termination of the loading procedure. The last entry in the table of external references is eight blank characters. Additional entries may be made to the table of external references by reassembling the table of external references and placing the assembled deck in the LLMPs system. The BPS assembler assigns each external symbol a unique ID ranging from 2 through 15 thus limiting a program to 14 external references. The OS/360 assembler allows a maximum of 255 external references in a program. The LOAD program will load only a job program that has 14 or less external references.

Changing Buffer Size:

The size of the buffer into which the LOAD program places the assembled LLMPs job program is 2048 bytes. If the length of the assembled program exceeds 2048 bytes, the LOAD program writes the message, 'ENLARGE BUFFER SIZE TO xxxxx' on the console typewriter. The size of the buffer required is designated by xxxxx. Use the CHANGBUF program to change buffer number 1 of the LOAD program as follows:

CHANGBUF LOAD SIZE xxxxx BFNO 1

System Routines:

The system card unit check routines CDRDUC, CDSTUC, and the system conversion routines BTOHEX and HXTOBX are used. The LOAD program uses three tables of external references with DC statements that contain addresses of system subroutines (described under external references). The tables are EXTAB1, EXTAB2, and EXTAB3. These tables are resident in the LLMPs system.

Task Request Summary:

LOAD RDRr

Task Request Format:

Jobname: LOAD

Units: RDRr

where r is the address of a 2540 card reader. If there is only one reader attached to the system, the reader unit is not specified.

Parameters: None

Operating Instructions:

The LOAD program must not be used to load an assembled LLMPs job program or joblist program which is resident in core as part of the current LLMPs system. The LOAD program is part of the current LLMPs system; it should be used only to load assembled job programs and joblist programs that are in the testing stage or that are used infrequently.

The LOAD program cannot be terminated using the TERM job. The UNLOAD job must be used to remove the assembled LLMPs job program and joblist programs placed in core via the LOAD program.

Because LOAD reads until the reader is empty the user of the LOAD program must place one assembled job program followed by the joblist(s) into the reader. The user must not stack jobs in the card reader. A maximum of six joblist programs may be loaded with one assembled job program. The user makes the reader ready and then initiates the LOAD program via the console typewriter.

A temporary restriction in the LOAD program requires the user to assemble separately the multiple joblists to a job program.

Console Messages: († Indicates that the task terminates)

1. 'xxxxxx AT LOCATION zzzzzz'

The LOAD program prints the program segment and entry point names with their storage addresses on the console typewriter. xxxxxx indicates a program name and zzzzzz its address.

2. PROGRAM LOADED

Indicates that the job program and joblist program(s) have been successfully loaded in core. The loaded job program may be initiated.

3. † UNIDENTIFIABLE CARD READ

Indicates that a card other than an ESD, TXT, RLD, END or REP card was read by the program. The task terminates.

4. † INVALID ESD EXTENDED CARD CODE

The assembler assigns a code to an external symbol dictionary card (ESD) identifying this card as a program name card, entry point card, or external symbol card. The assembler assigns a code of 0 to the program name card, a code of 1 to an entry point card, and a code of 2 to an external symbol card. Any other code is invalid and causes the task to terminate.

5. † ESD TYPE 0 NOT FIRST

The program name card designated as the ESD TYPE 0 card must be the first card of an object deck. If not, the task terminates.

6. † INVALID ESID

The assembler assigns an external symbol identification number of 1 to program name cards and entry point cards. On an external symbol card, the load program expects an ESID ranging from 2 to 15 and so there may not be more than 14 external references in any given program segment. The task terminates.

7. † EXTERNAL REFERENCE NOT IN TABLE

The external symbol card reference is not contained in the system tables of allowable external references. The task terminates.

8. † TXT ADDRESS BELOW BUFFER

The address in storage where the information on the text card is to be loaded is below the buffer start address. The task terminates.

9. † TXT ADDRESS EXTENDS BEYOND BUFFER
The address in storage where the information on the text card is to be loaded is beyond the buffer end address. The task terminates.
10. † ENLARGE BUFFER SIZE TO xxxxx
The buffer size of the load program must be increased to xxxxx bytes. The task terminates.
11. † ILLEGAL CHARACTER IN REPLACE CARD
The program has found a character other than hex digits 0-9, or A-F in a replace card. The task terminates.
12. † ODD ADDRESS IN REPLACE CARD
The starting address in hexadecimal of the area to be replaced with information contained on the REP card must be on a half word boundary. The task terminates.
13. † MORE THAN 11 FOUR-DIGIT FIELDS IN REPLACE CARD
A replace card can have a maximum of 11 four-digit hexadecimal fields of corrections. Each field is separated from the previous field by a comma and replaces one previously loaded half-word. If more than eleven fields are present, the task terminates.
14. † JOBLIST(S) MISSING
No joblist has been read by the LOAD program. The task terminates.
15. † PROGRAM ENTRY NOT FOUND FOR JOBLIST EXTERNAL REFERENCE
The external reference made by the loaded joblist does not correspond to the loaded program segment or its entry points. The joblist is illegal and the task terminates.
16. † JOBLIST PROGRAM REPLACE CARD ADDRESS INVALID
The address specified on the replace card for a joblist program is invalid. The task terminates.
17. † JOBLIST TXT EXCEEDS JOBLIST BUFFER
The JOBLIST may contain only 16 bytes of text information. The task terminates.
18. † ONLY x JOBLIST(S) MAY BE LOADED AT PRESENT.
There is room for only x joblists in core. The task terminates.

19. † MORE THAN SIX JOBLISTS

The LOAD program only allows six joblists to be associated with the job program which it is loading. The task terminates.

20. † DUMMY JOBLIST NOT FOUND

A dummy joblist to be replaced in core by the loaded joblist is not available. The task terminates.

21. † DEVICE NOT FREED

The reader has not been freed once the load program has loaded the job program and joblists into core. The task terminates. If this condition persists, notify a system programmer.

Program Logic:

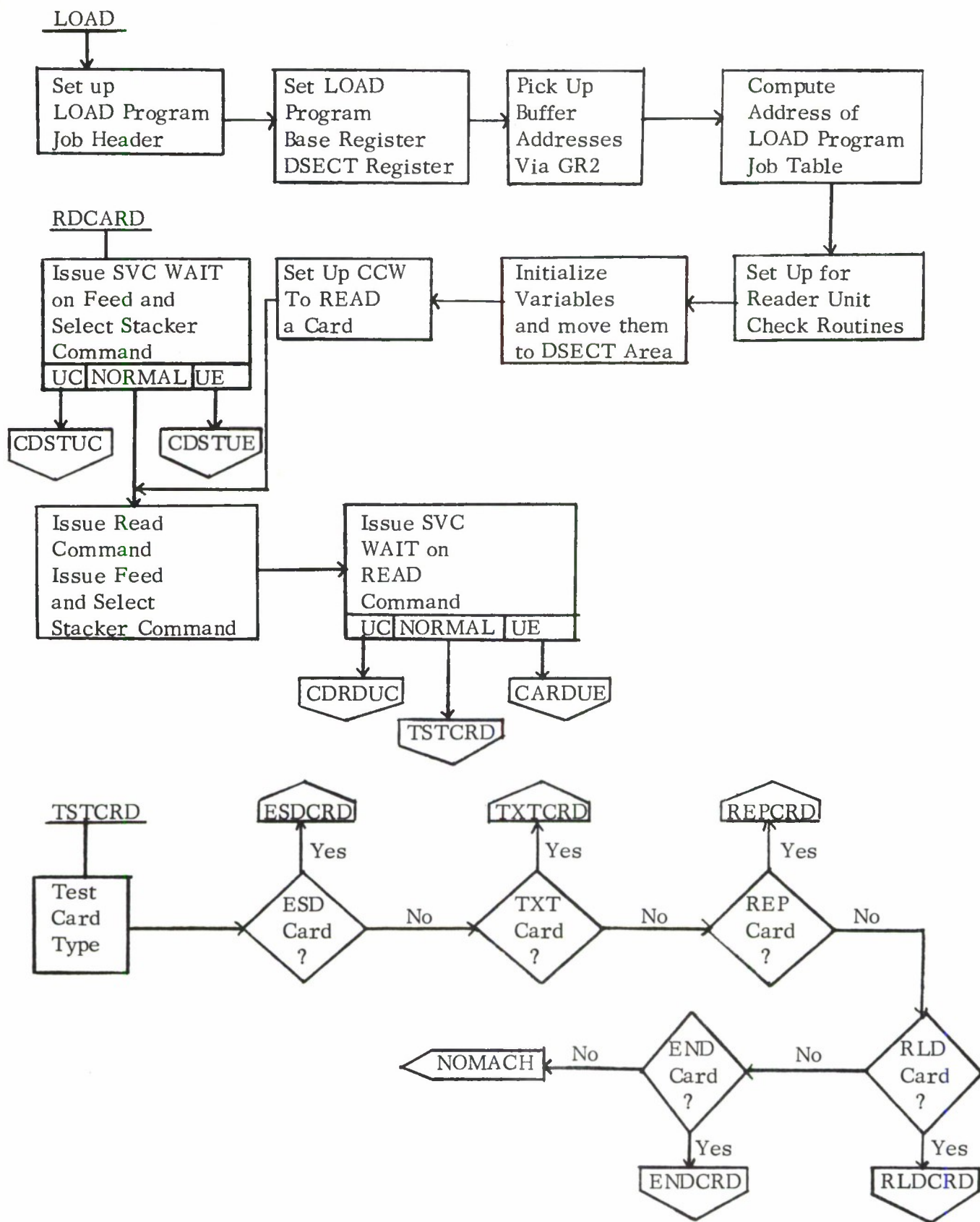
The LOAD program, at initiation, requests a buffer of 2048 bytes. Into this buffer, the LOAD program loads the assembled LLMPs program. After all cards have been read, the LOAD program searches through the joblist storage area of core for the DUMMY joblists. There are six dummy joblists resident in core as part of the LLMPs system. These dummy joblists are replaced with the loaded joblist programs. A maximum of six joblist programs may be loaded with one job program. If the LOAD program doesn't find a dummy joblist in core, a message, 'DUMMY JOBLIST NOT FOUND,' will appear on the console typewriter. If the LOAD program doesn't find enough dummy joblists to be replaced by the loaded joblists, the message, 'ONLY x JOBLISTS MAY BE LOADED AT PRESENT' will be printed on the console typewriter. In each case the job terminates.

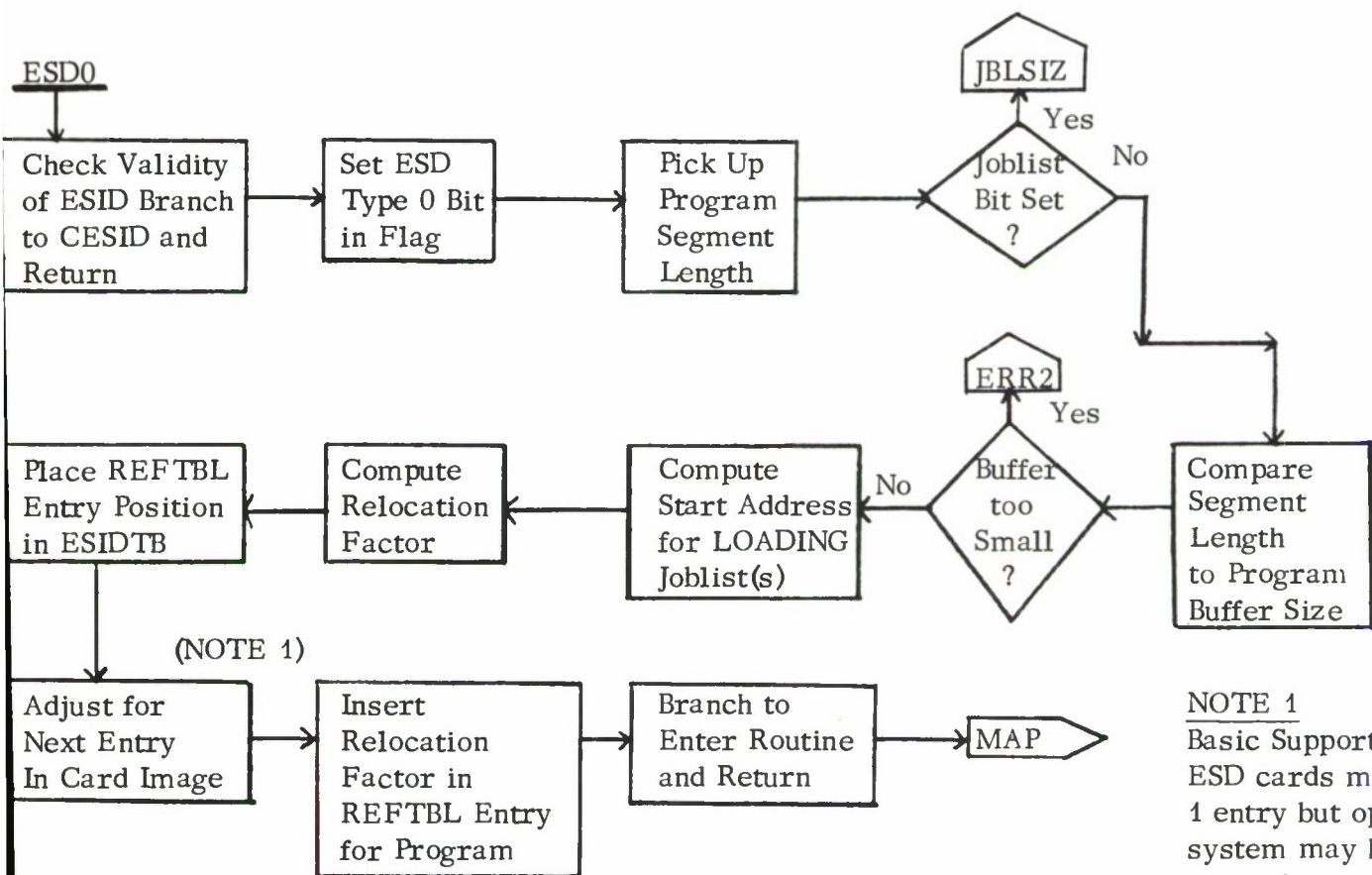
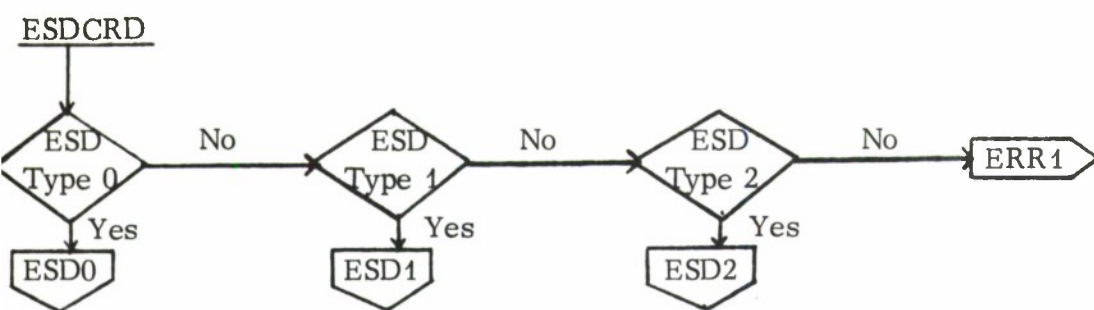
A description of the 16 byte dummy joblist entry follows:

DUMLST	START	0
	EXTRN	UNLOAD
	DC	C'DUMMYLST'
	DC	C'1'
	DC	x'00'
	DC	H'0000'
	DC	A(UNLOAD)
	END	

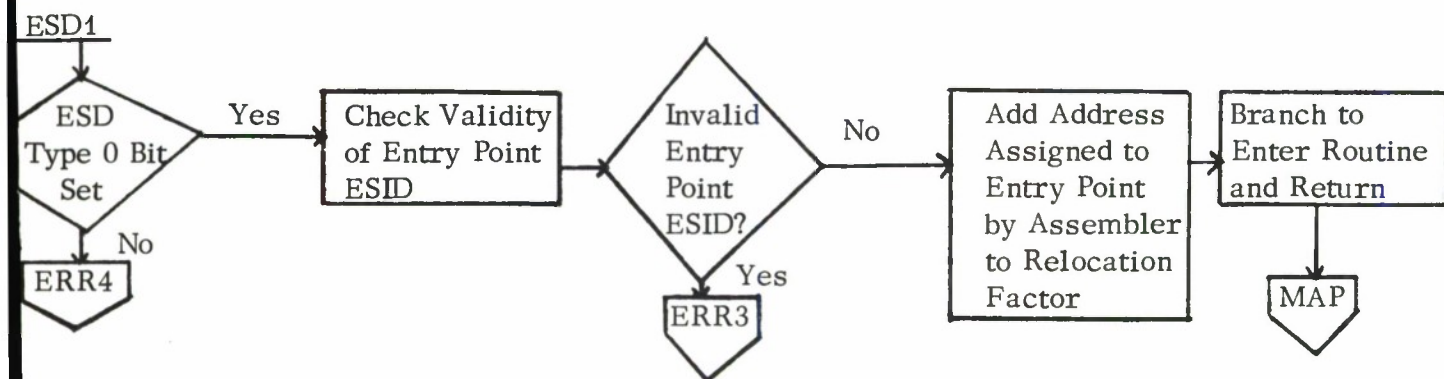
When loading is complete, the LOAD program writes a message, 'PROGRAM LOADED', to the operator and then issues an SVC FREE to release the reader device. The LOAD program then issues an SVC POPQ to remove the LOAD job from the CPU QUEUE. Such action relinquishes control of the central processing unit without terminating the job. If the reader device cannot be freed, the LOAD program will write a message on the console typewriter and will terminate. Since the LOAD job does not normally terminate after the loading function, its buffer is not cleared.

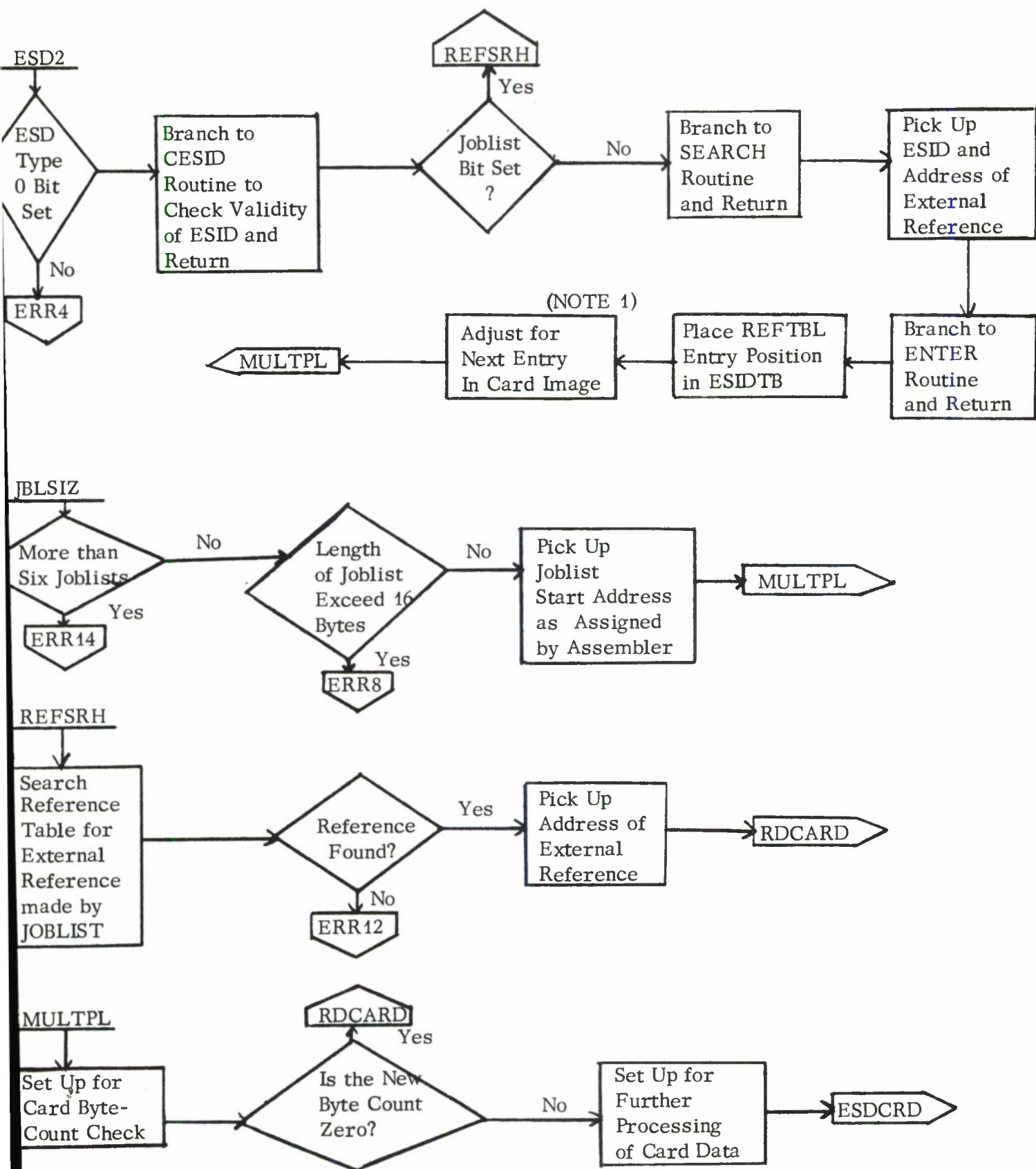
The overall structure of the LOAD program is described in the flow charts that follow:

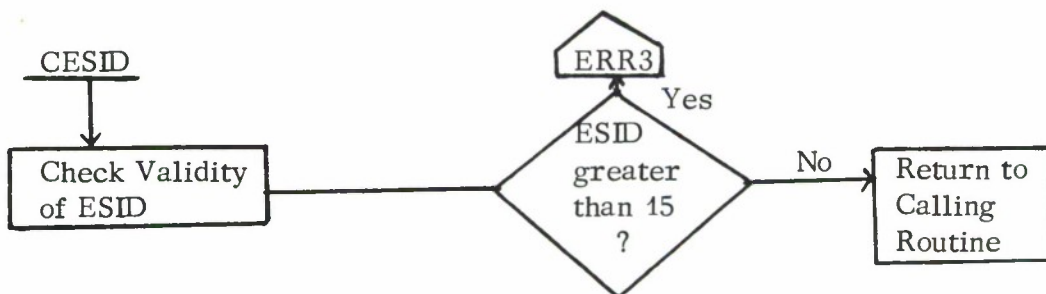
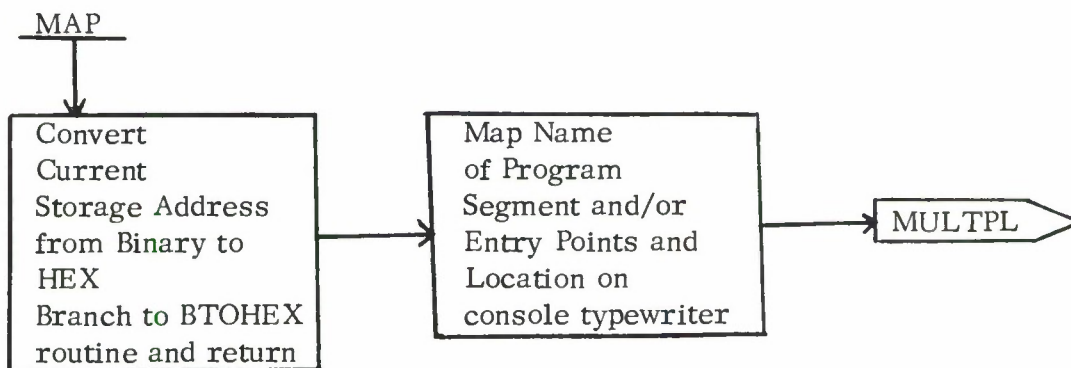
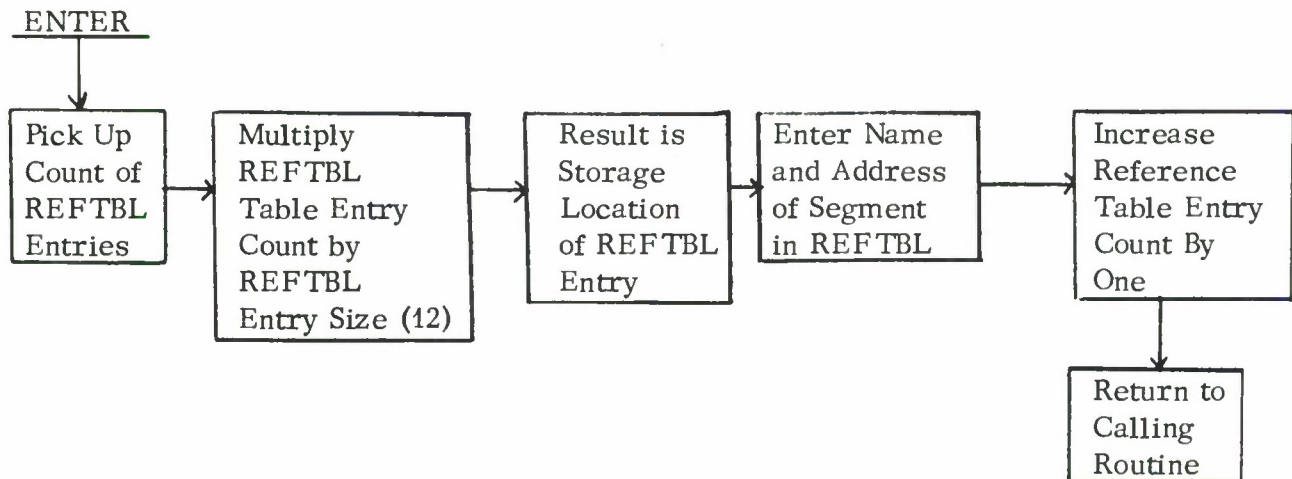


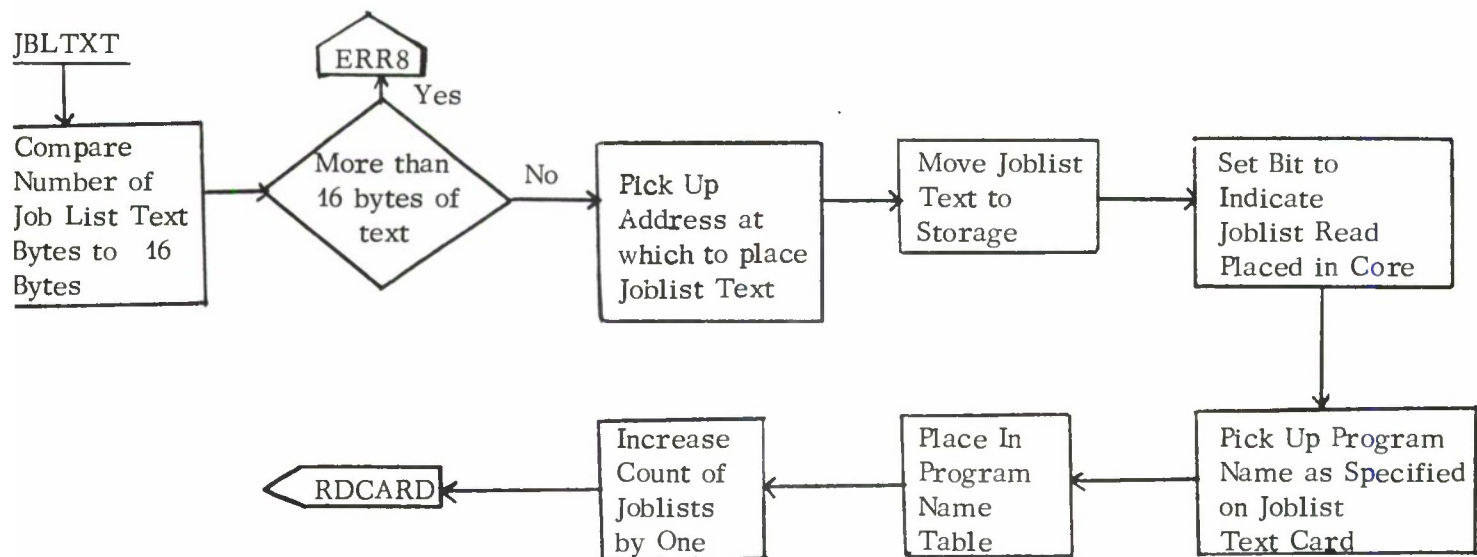
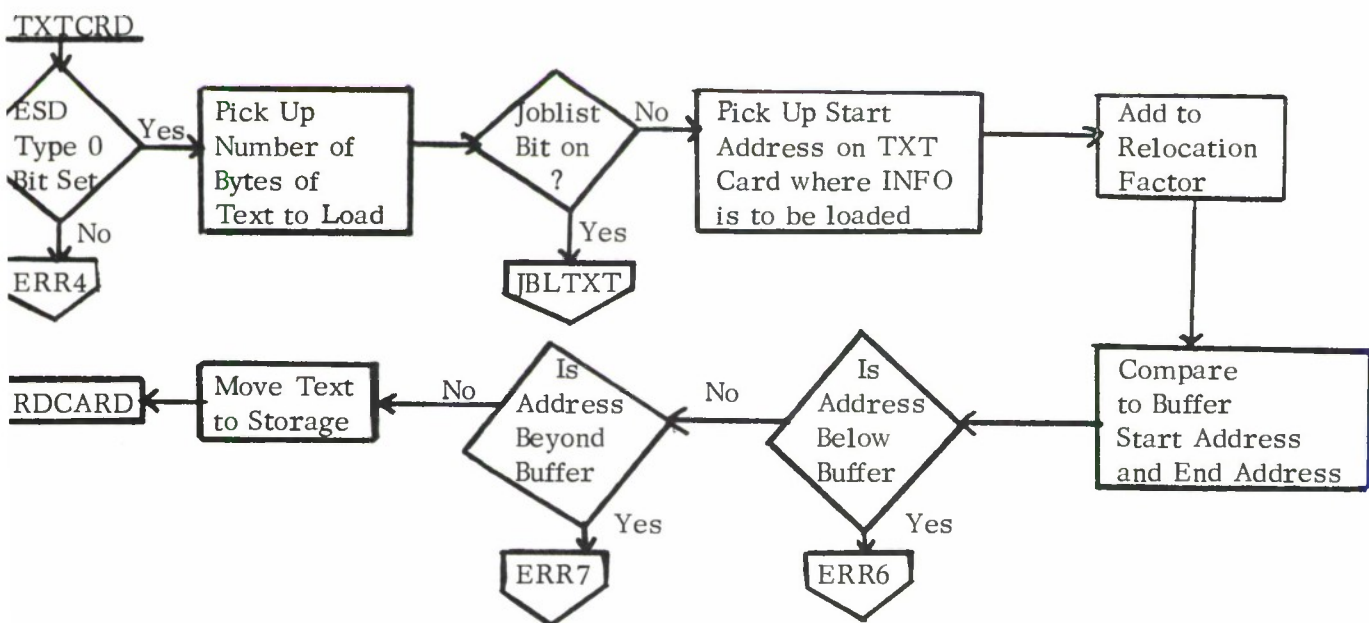
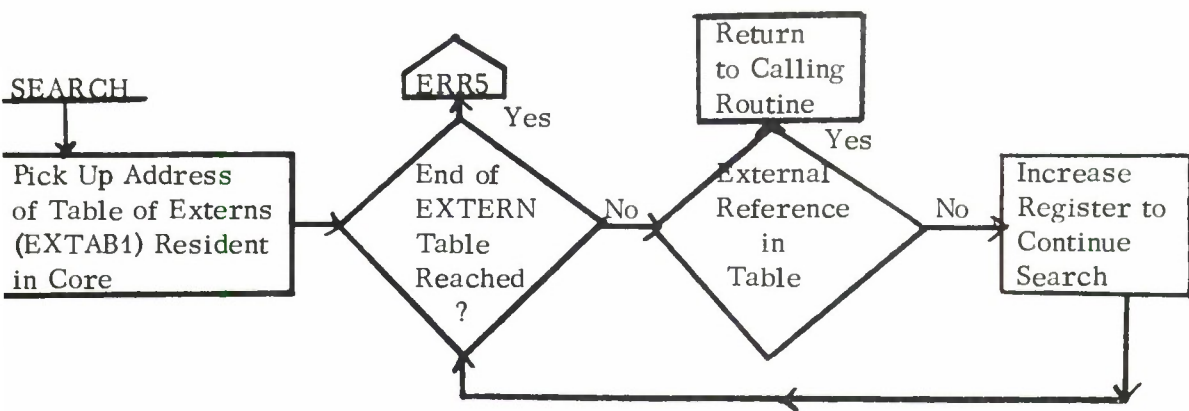


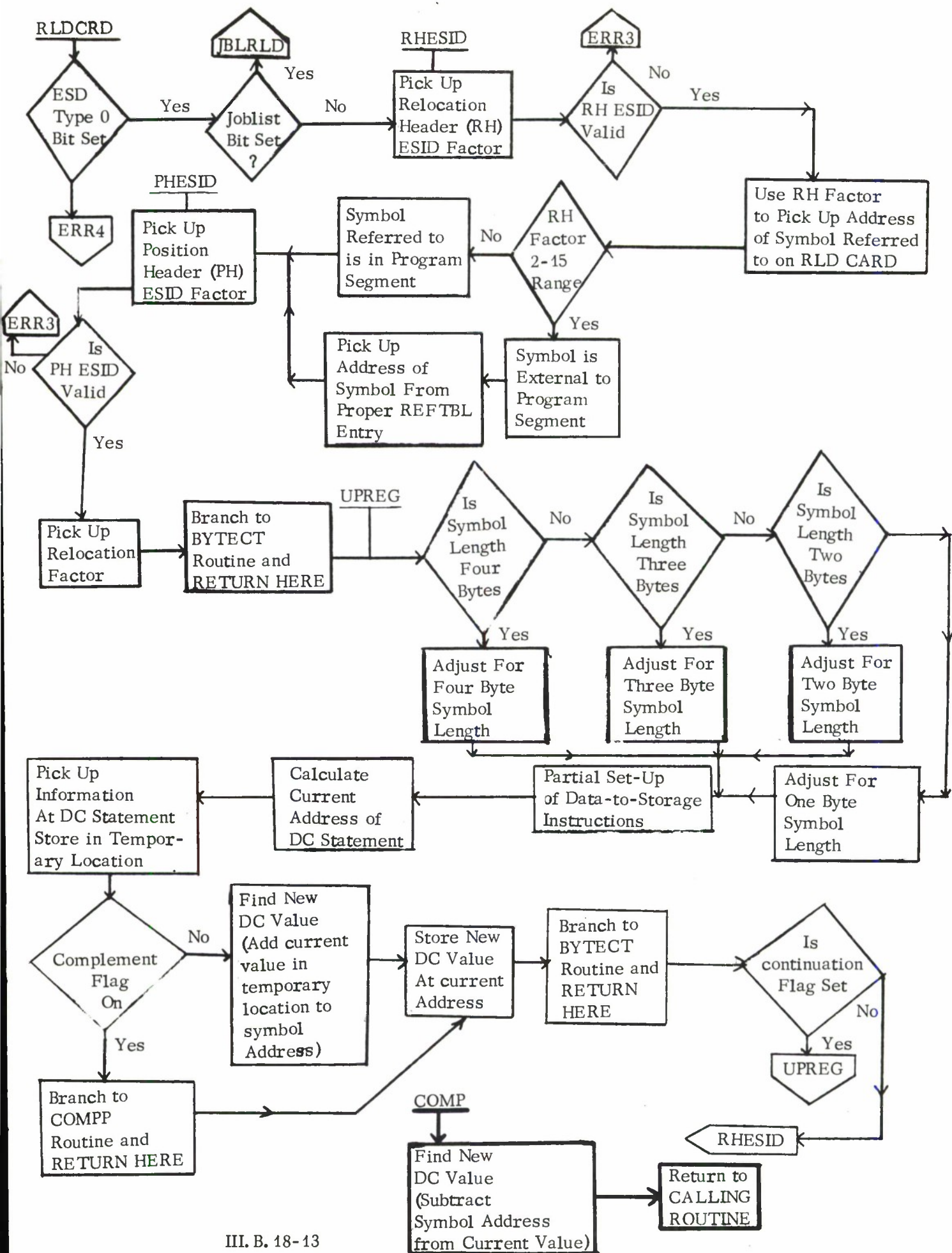
NOTE 1
Basic Support
ESD cards may have
1 entry but operating
system may have
more than one.

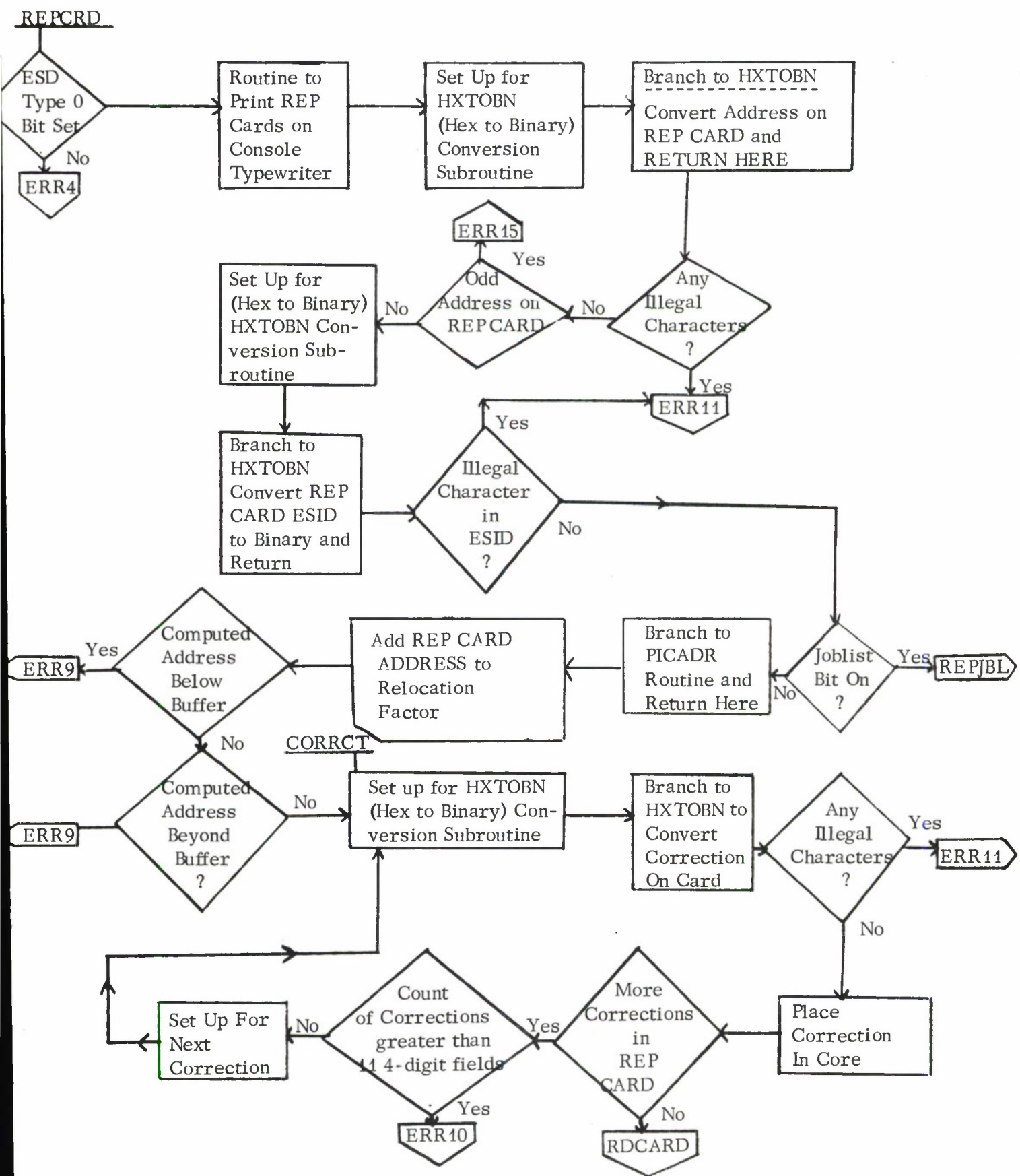


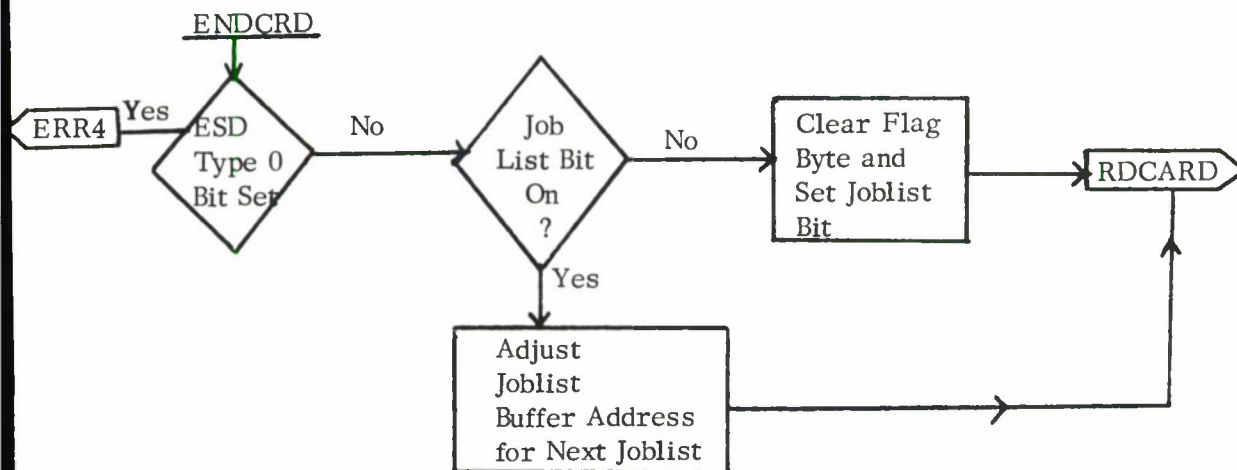
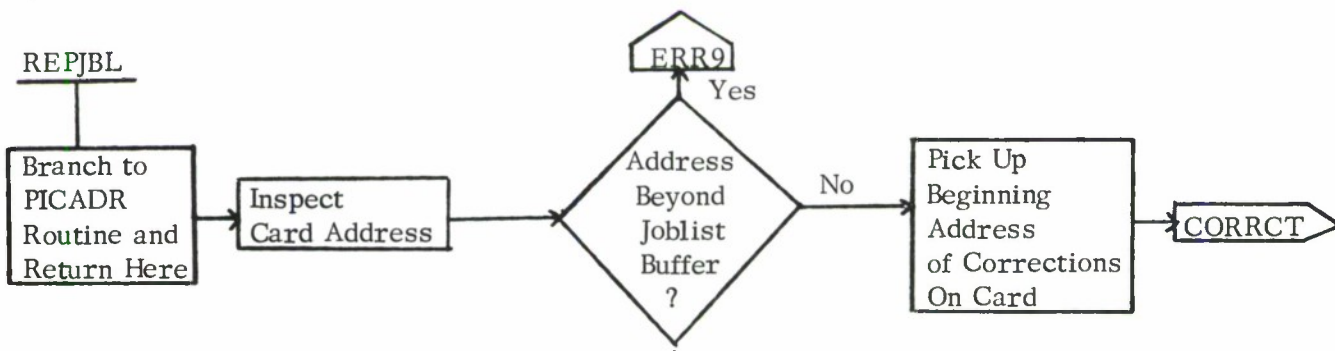
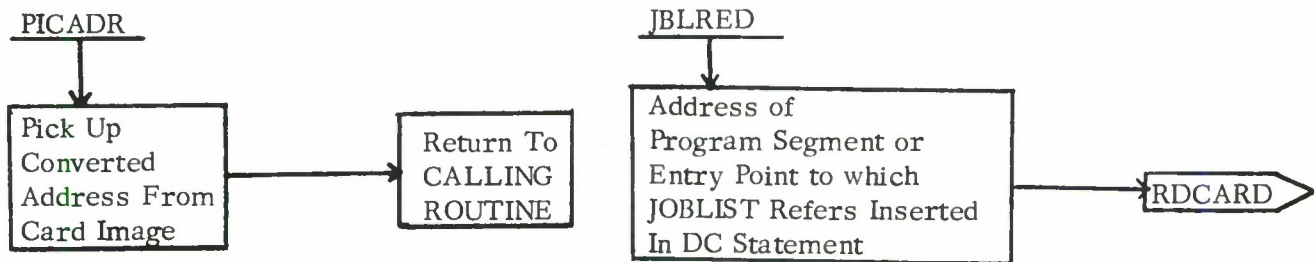


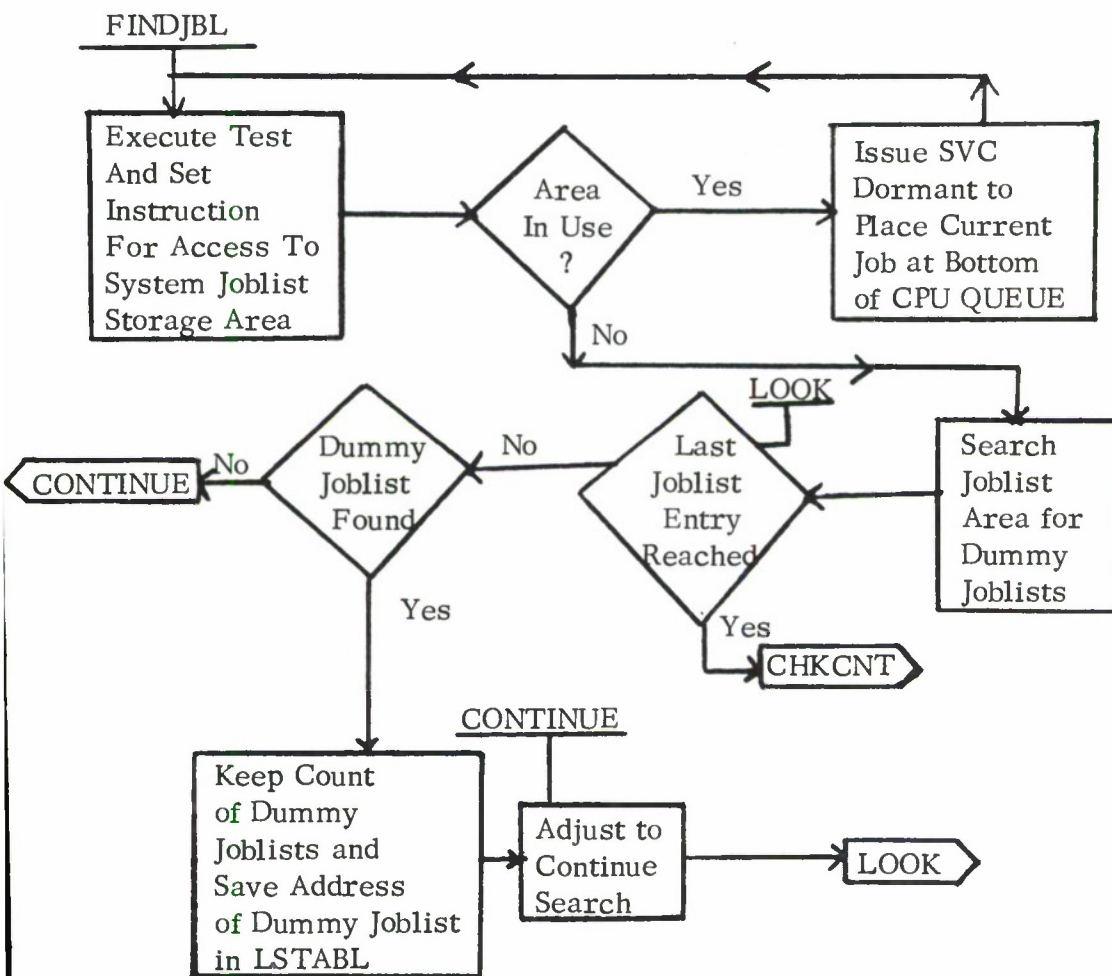


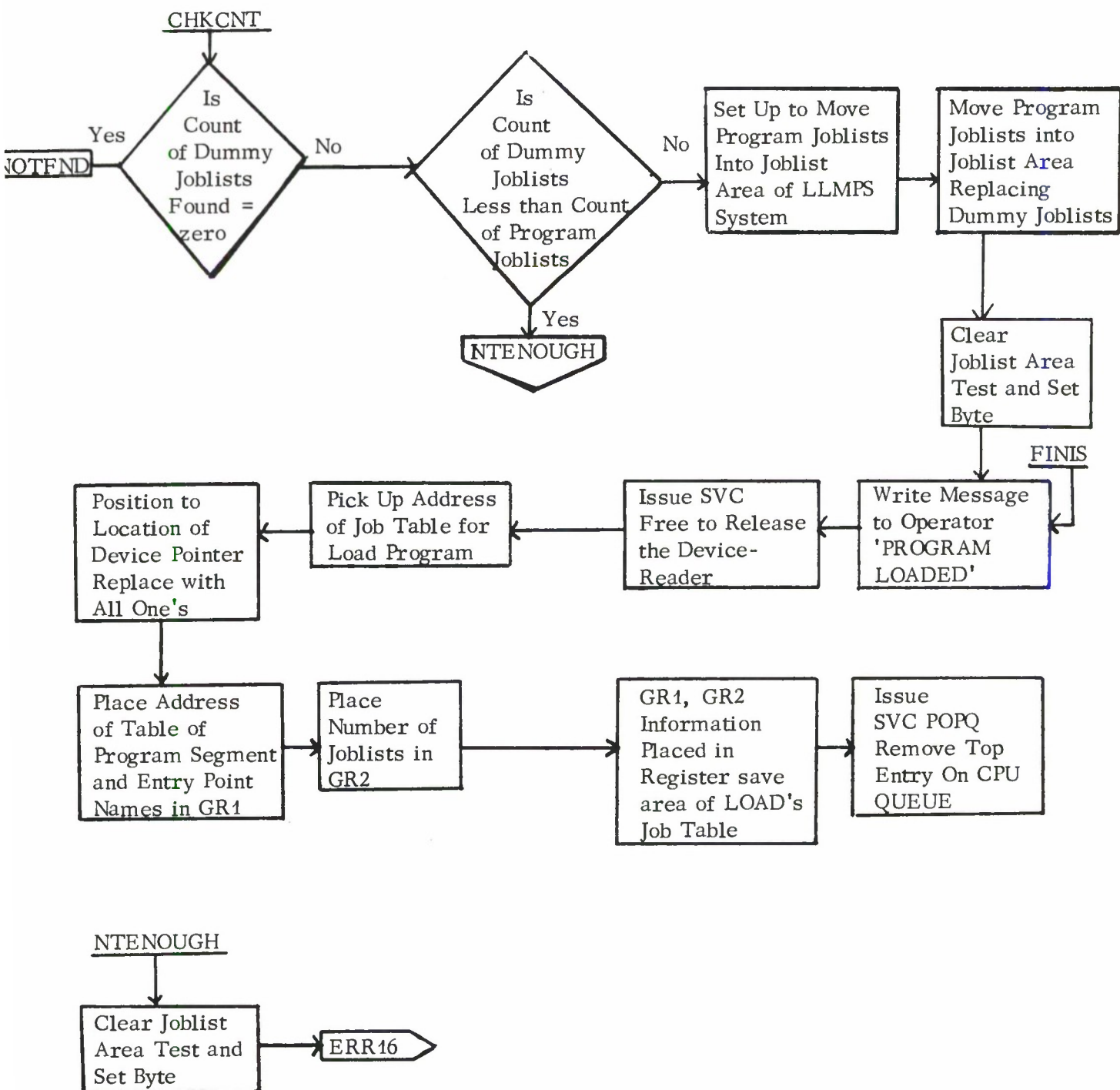


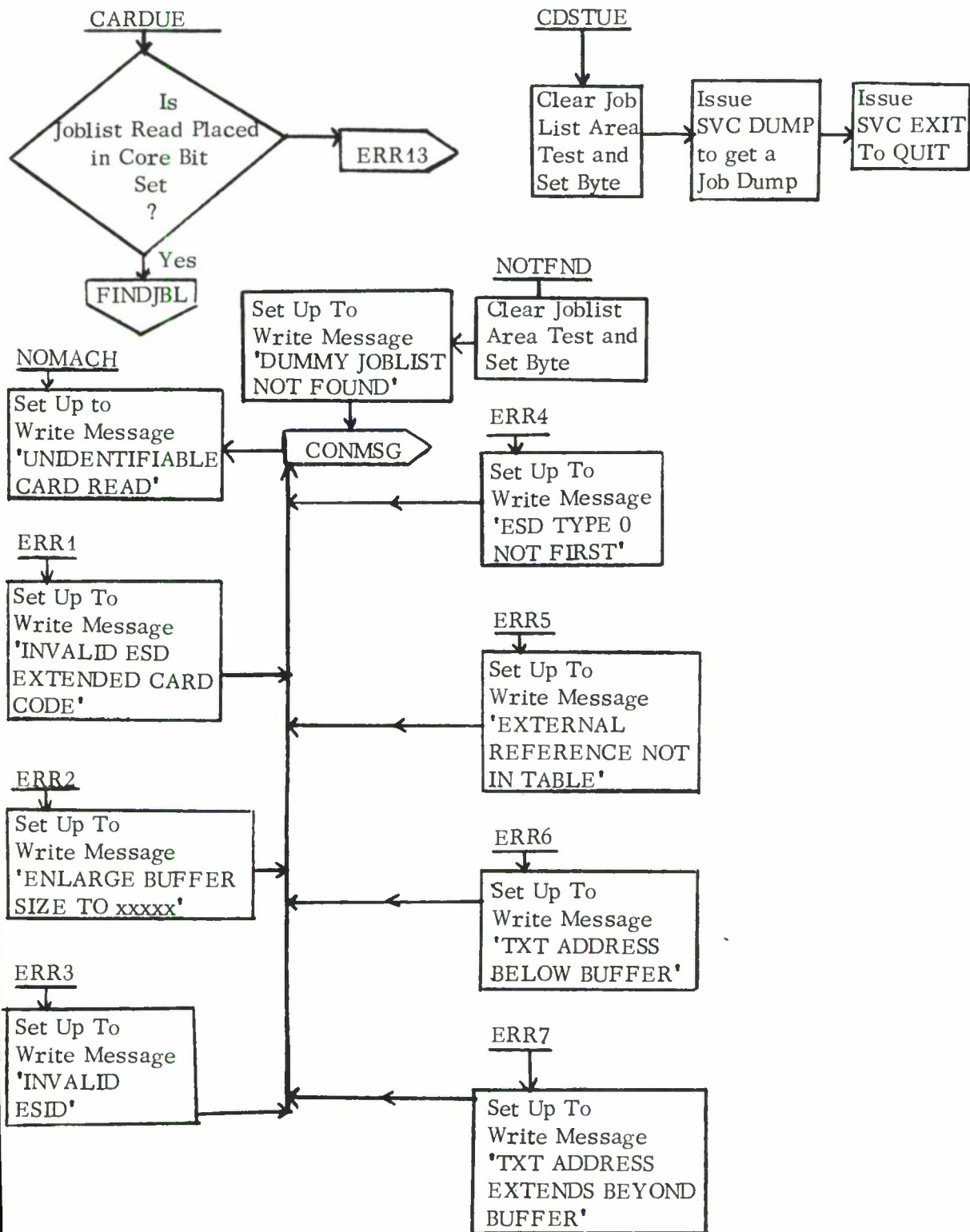


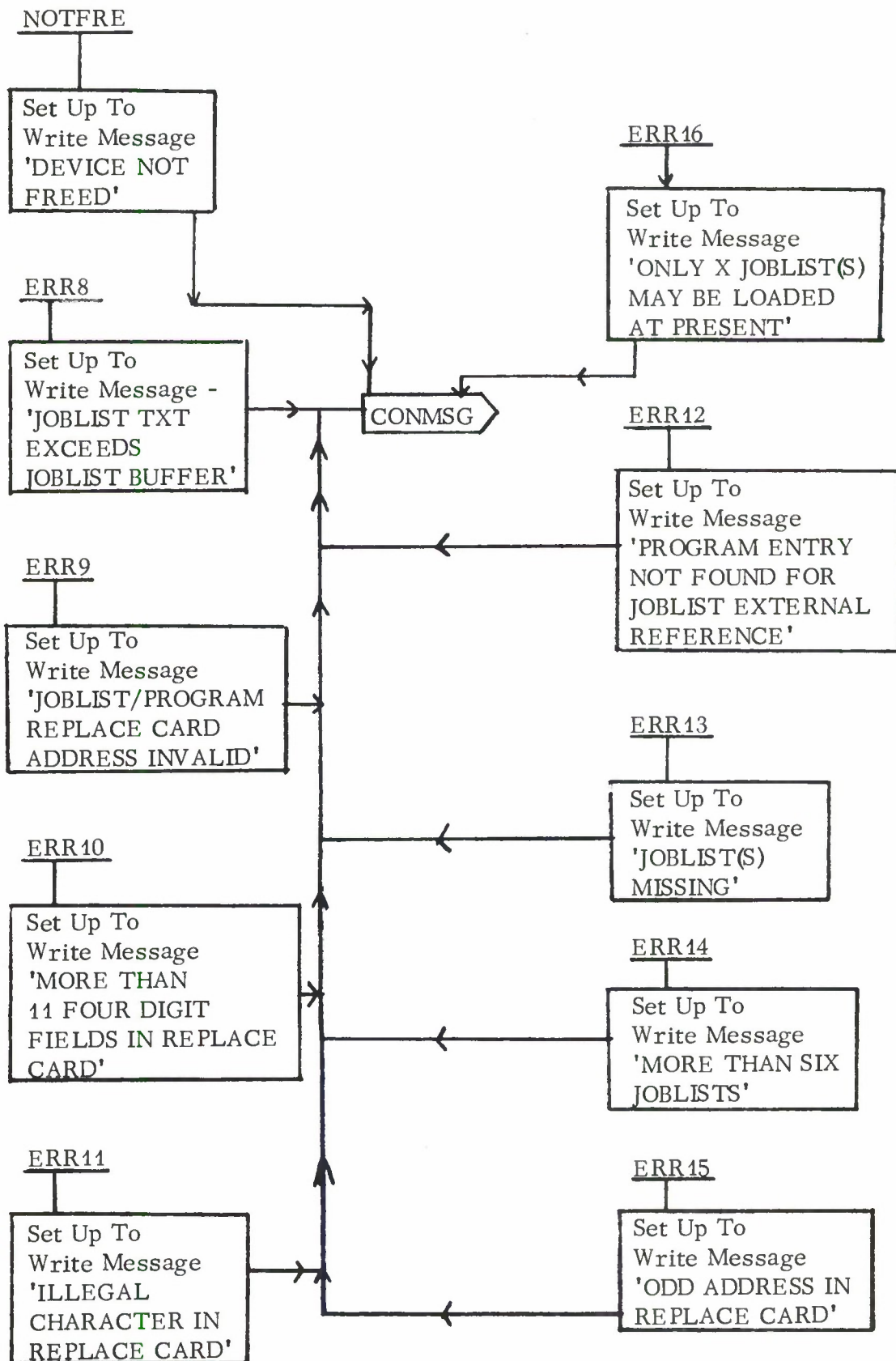












UNLOAD

Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

UNLOAD Program (Re-entrant)

20 March 1967

Job Summary:

The UNLOAD program removes from core storage the job programs and joblist programs loaded into core by the LOAD program.

General Information:

The UNLOAD program unloads job programs and joblist programs previously loaded by the LOAD program. The UNLOAD program replaces the joblists with the DUMMY joblists. With a branch to the system routine FBJTJL, the UNLOAD program frees the buffers, job table, and joblist of the LOAD task that loaded the job program and joblist programs into core.

The system routine FBJTJL is used by the UNLOAD program.

Task Request Summary:

UNLOAD

Task Request Format:

Jobname: UNLOAD
Units: No devices used.
Parameters: None

Operating Instructions:

Upon initiation, the UNLOAD program will type a message on the console type-writer requesting the operator to type the name of the program to be unloaded. The operator should then type the name of the program to be unloaded. For unloading the TPCHECK job, the console listing would look as follows:

```
0001 UNLOAD OPERATOR TYPE JOBNAME  
0001 UNLOAD ... tpcheck
```

The UNLOAD job unloads every job program contained within the same control section as the job program whose name is supplied by the operator, provided that none of these jobs is active. If the jobname typed or any of the other jobs in the control section is active, the UNLOAD program will not perform its function and will terminate.

Console Messages: († indicates task terminates)

1. OPERATOR TYPE JOBNAME

The UNLOAD program requests the operator to type the name of the program to be unloaded. The UNLOAD program then issues an SVC READ to obtain the operator's reply.

2. † xxxxxxxx UNLOADED

The UNLOAD program lists the programs unloaded (designated by xxxxxxxx) and then terminates.

3. † xxxxxxxx NOT LOADED BY LOAD PROGRAM

The LOAD program did not place the job program in core storage. The name of the job program is designated by xxxxxxxx. The task terminates.

4. † xxxxxxxx IS ACTIVE

The program (designated by xxxxxxxx) is active and therefore must not be unloaded at present. The task terminates.

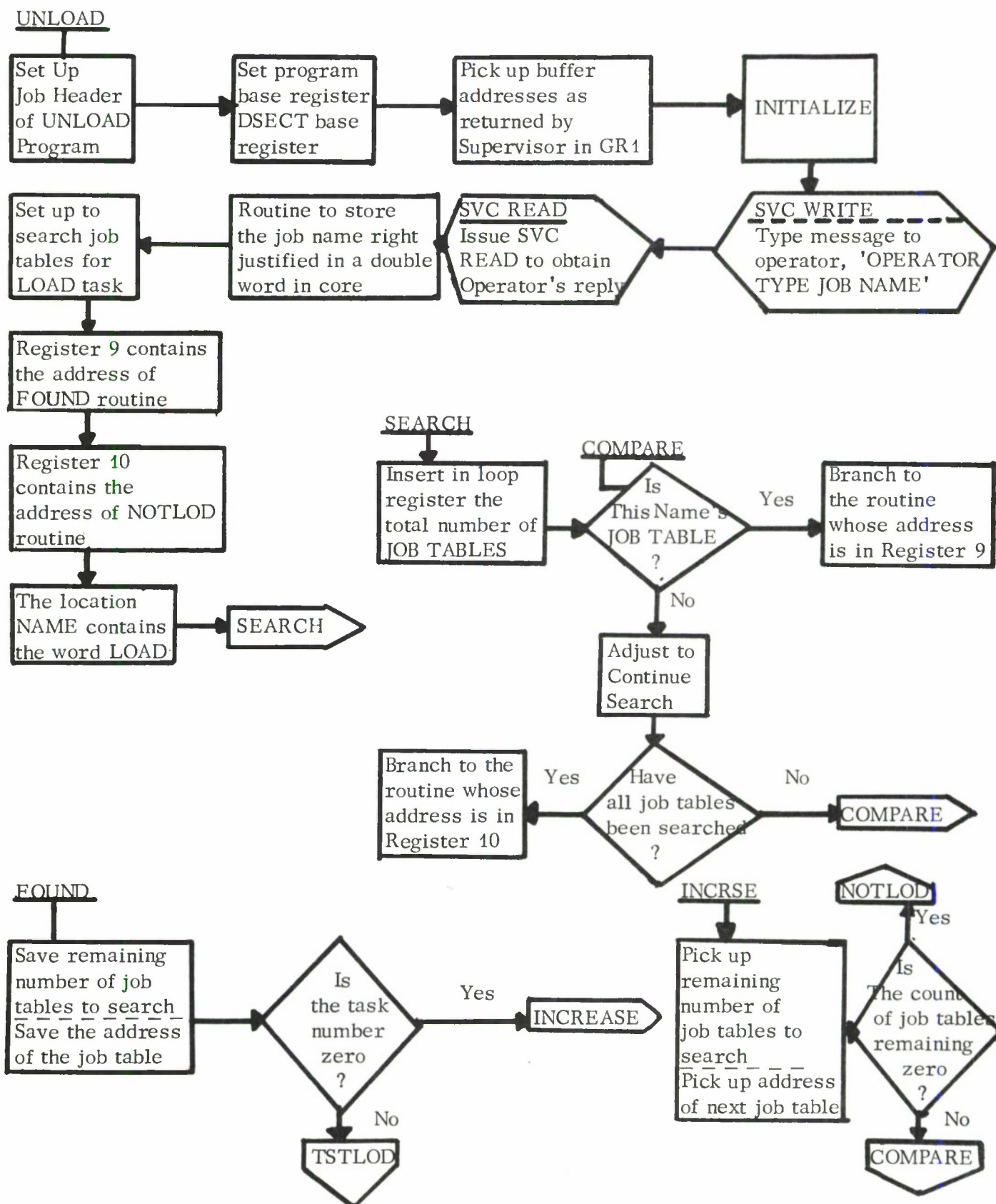
5. † CATASTROPHIC ERROR NOTIFY DIANA

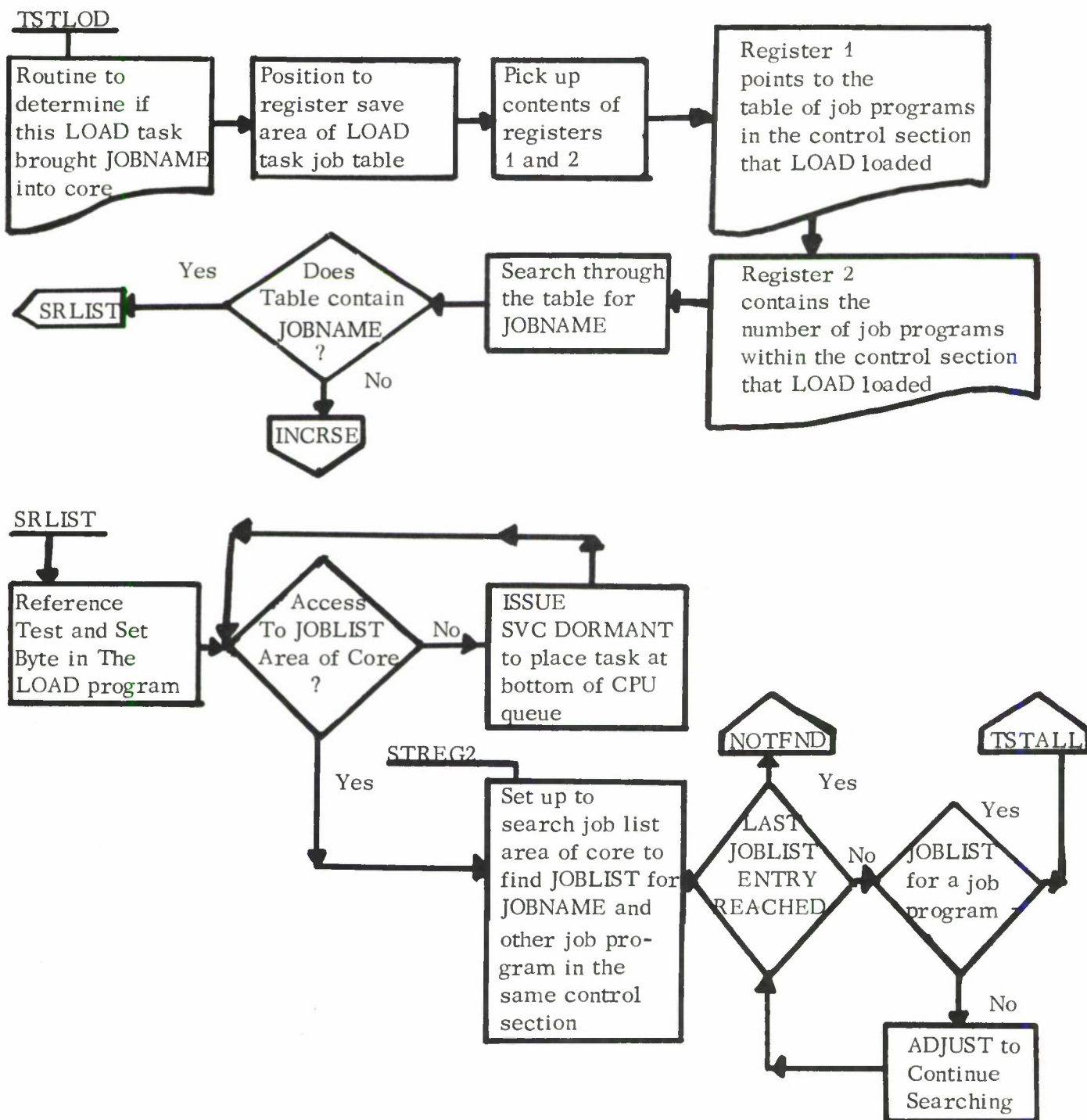
A job list entry can't be found in the job list area of core storage for a program loaded via LOAD. This is an impossible situation. The task terminates.

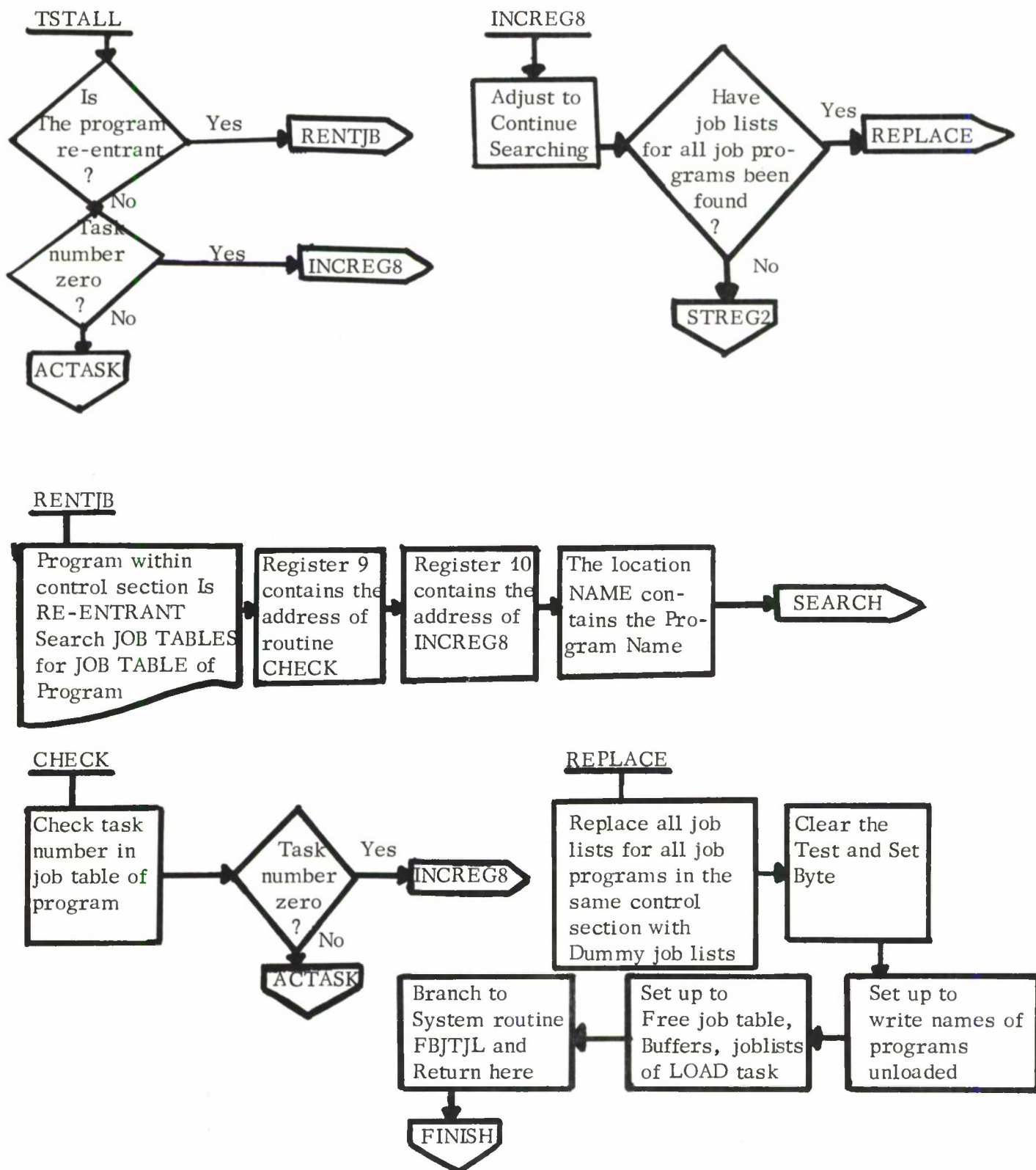
Program Logic:

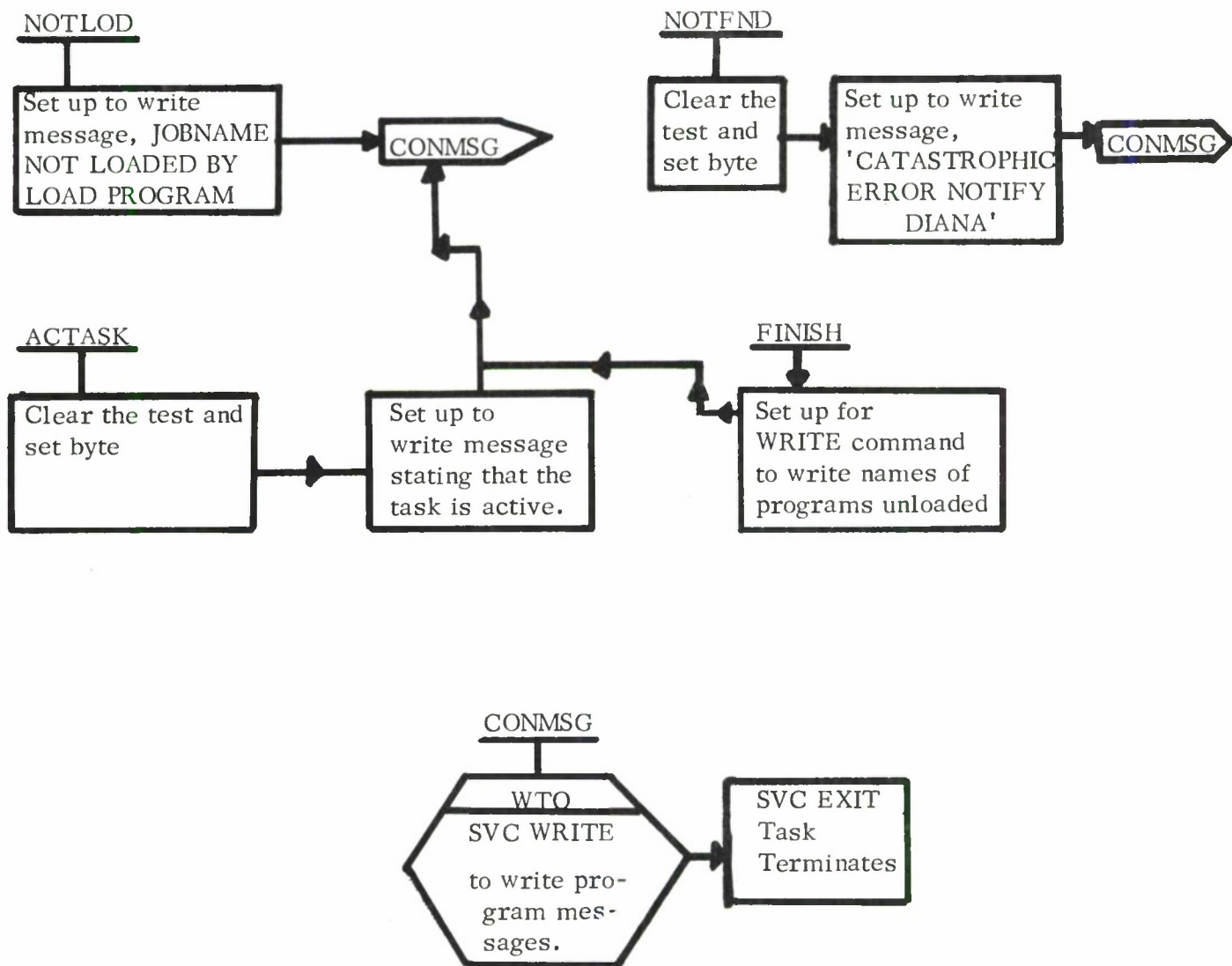
The LOAD task that loaded the job and joblist(s) is then terminated with the code UNLD. The UNLOAD program lists the names of the program(s) unloaded on the console typewriter and terminates. The structure of the UNLOAD program is revealed by the following flow diagrams.

Flow Diagram of UNLOAD Program









Lincoln Laboratory Multi-Programming Supervisor

Job Operating Instructions

SHARE Job

1 April 1967

Job Summary:

The SHARE job will list and punch catalog and/or order cards from a SHARE Distribution Tape Reel (DTR) or will retrieve source and object decks from the DTR or from a SHARE Series Library Tape (SLT).

General Information:

The SHARE DTR and SLT tapes are assumed to have the SHARE Distribution Tape Reel format and the SHARE Library Tape format as of December 12, 1966. These formats are described in program write-ups from the IBM 1401 General Program Library, file numbers 1401-13.1.026, 1401-13.1.028 and 1401-13.1.032.

SHARE uses the system subroutines TPRDUC, PRTUC, PRTUE, CDRDUC, CDSTUC and CDPNUC.

Job Request Summary:

SHARE RDRr PCHc PTRp Txxx JOB n

DEN2
DEN5
(DEN8)

Task Request Summary: JOBNAME UNITS PARAMETERS

Jobname: SHARE

Units: RDRr PCHc PTRp Txxx

where r is the address of a card reader, c is the address of a card punch, p is the address of a printer, and xxx is the address of a 7-track tape. If there is only one reader, punch or printer attached to the system, the addresses of the respective units need not be specified. All four devices must be assigned to the task even though each job requires only three devices.

NOTE: The reader and punch must be on the same 2540 unit because jobs 4 and 5 interleave order cards from the reader with punched cards from the punch into stacker RP3.

Parameters:

Density:	DEN2	(200 bpi)	Default: DEN8
	DEN5	(556 bpi)	
	DEN8	(800 bpi)	
Job:	JOB	n	Default: Parameter required

where n may be any one of the following
job request numbers depending on the
task desired:

- 1-list and punch catalog and order cards
- 2-list and punch catalog cards
- 3-list and punch order cards
- 4-tape to card from a SHARE DTR
- 5-tape to card from a SHARE SLT

Operating Instructions:

For job requests 4 and 5, order cards placed in ascending order according to the deck number appearing in cols. 21-24, are read from the card reader. If the cards are not in ascending order, a message will appear on the console typewriter and the job terminated. Each order card followed by a blank card precedes its corresponding deck into stacker RP3. For jobs 1, 2, and 3 punched cards are also fed into stacker RP3 with a blank card inserted as the first card of each deck.

Only one job may be requested at a time and for jobs 1, 2, 3 and 4 the tape must be positioned at load point at job initiation.

Examples:

1. SHARE RDRA PCHB PTRE T181 JOB 1

The catalog and order cards on tape 181 will be listed on printer E and punched on punch B. Even though the reader is not used for this job, a reader unit must be assigned.

2. SHARE RDRC PCHD PTRF T182 DEN5 JOB 5

The tape on 182 is read at 556 bpi. The source and object decks requested by the order cards placed in reader C are punched on punch D. A printer is not required for this job, but a printer unit must be assigned.

Console Messages: († implies task terminates)

1. †PARAMETER ERROR

An erroneous parameter was typed.

2. † NO JOB SPECIFIED

The job parameter was not supplied.

3. PARITY

A tape parity error occurred. When a tape parity error is detected, the tape is reread 15 times. If the error still persists, the above message is typed on the console typewriter, the record is read with error, listed and/or punched, and the job continues. No indication of the error condition appears on the listing or the punched card.

4. † CARDS NOT IN ASCENDING ORDER

The deck number on tape is greater than the deck number on an order card; therefore, either the order cards were not placed in ascending order in the reader or the deck requested is not on the tape.

5. † END OF JOB

The job was successfully completed.

6. † x TAPE IMPROPERLY POSITIONED

Where x is an error code indicating the reason the program considers the tape improperly positioned. The error codes are:

- 1--An unexpected end of file occurred while attempting to skip tape the appropriate number of records, corresponding to the particular job request. (See Program Logic).
- 2--The physical record length was greater than 1600 bytes. No records on tape (except those skipped while properly positioning the tape) should be greater than 1600 bytes.
- 3--The physical record length was less than 80 bytes. No records on tape should be less than 80 bytes.
- 4--A record on the DTR which was to be listed and punched was not written in even parity.

The above message indicates that a possible program error exists, that the input tape was improperly written or positioned, or that the wrong input tape is mounted. The program requests a job Dump before terminating.

7. † PUNCH UNIT EXCEPTION

Unit exception on the punch, feed, select stacker command indicates a machine error. Notify CE.

8. † FEED UE

Unit exception on a feed command indicates machine error. Notify CE.

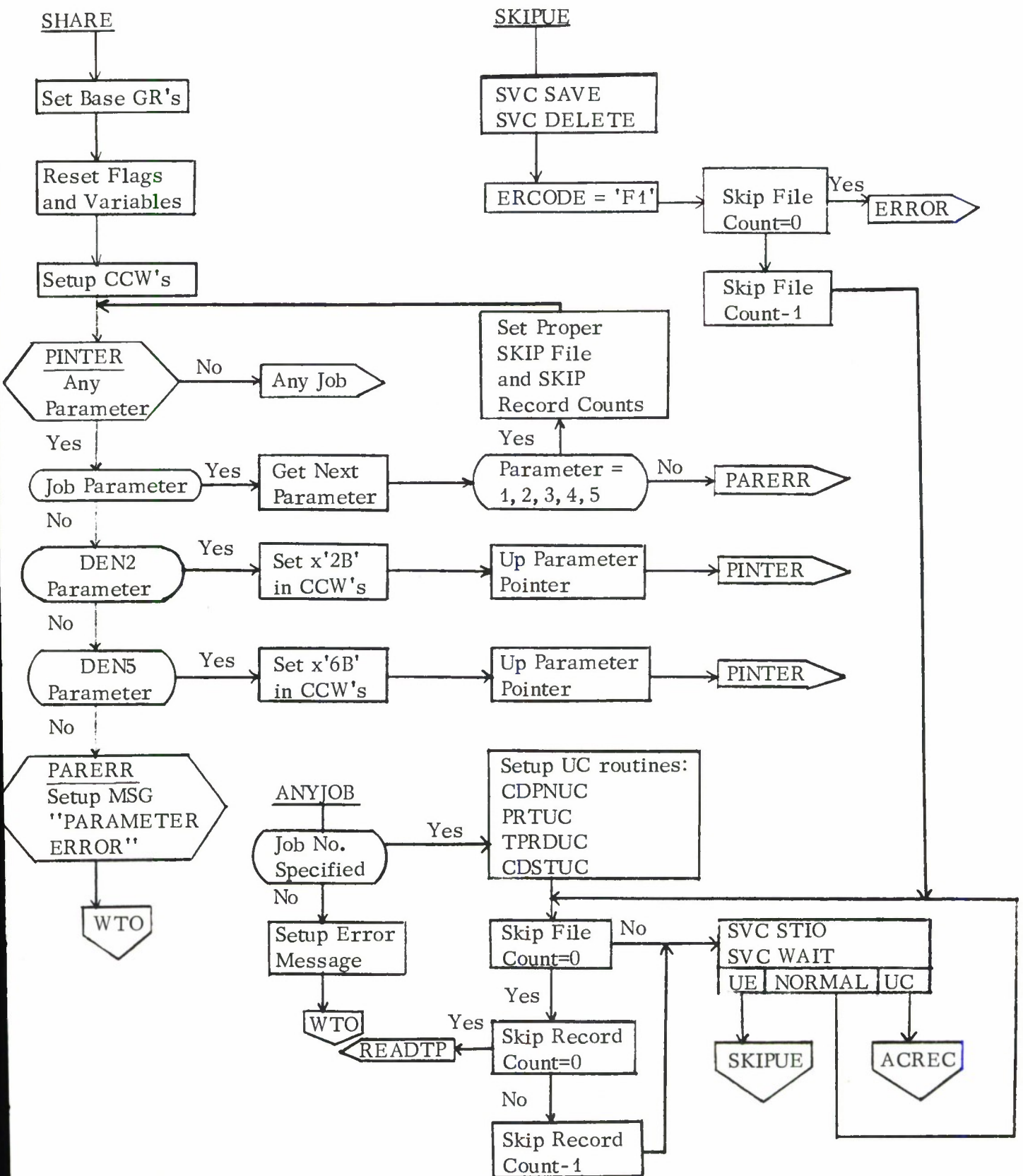
Program Logic:

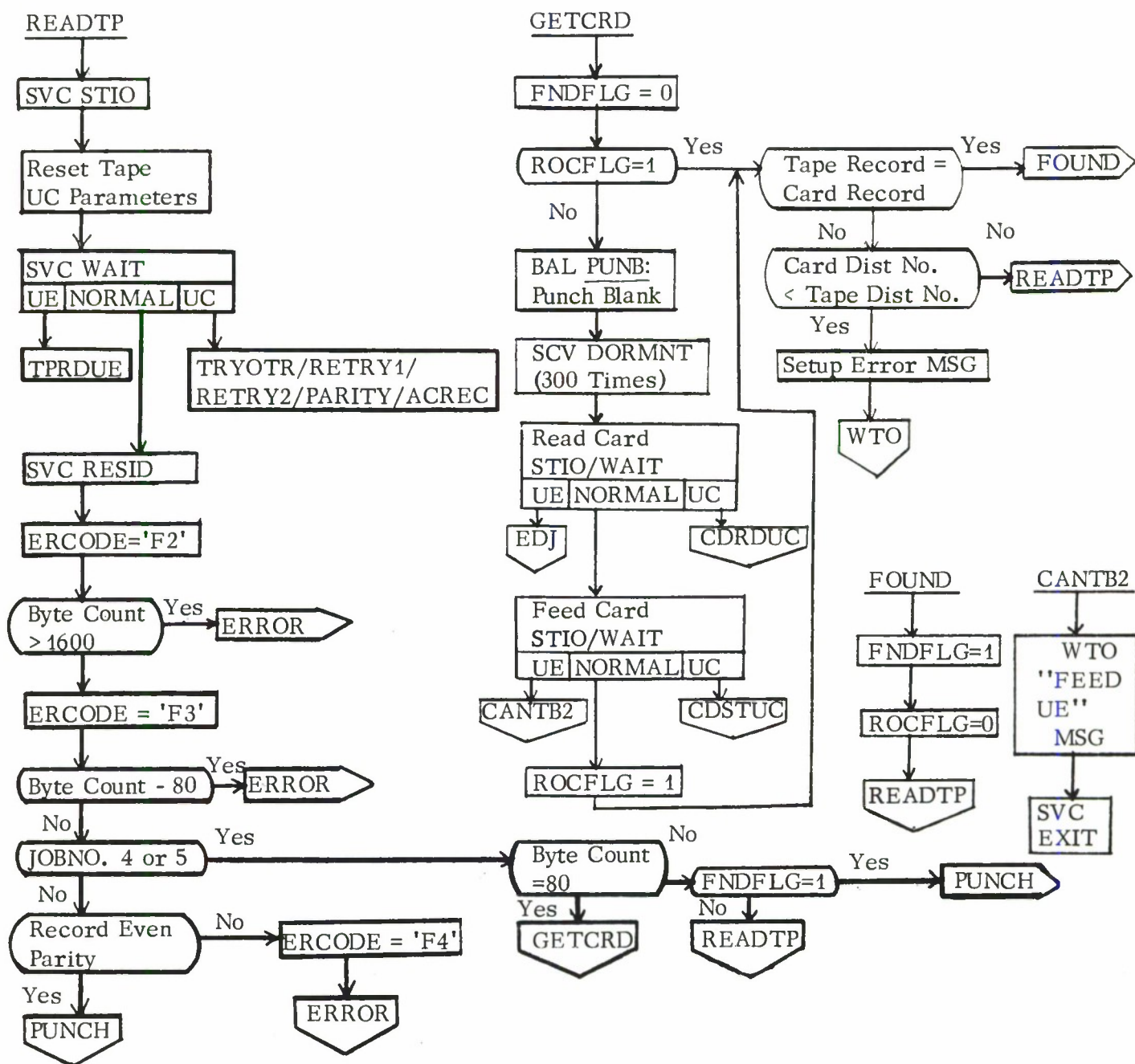
Because of the particular format of the DTR tape, the program upon initiation and before tape processing, will forward space a DTR tape the following number of files and/or records depending on the job requested:

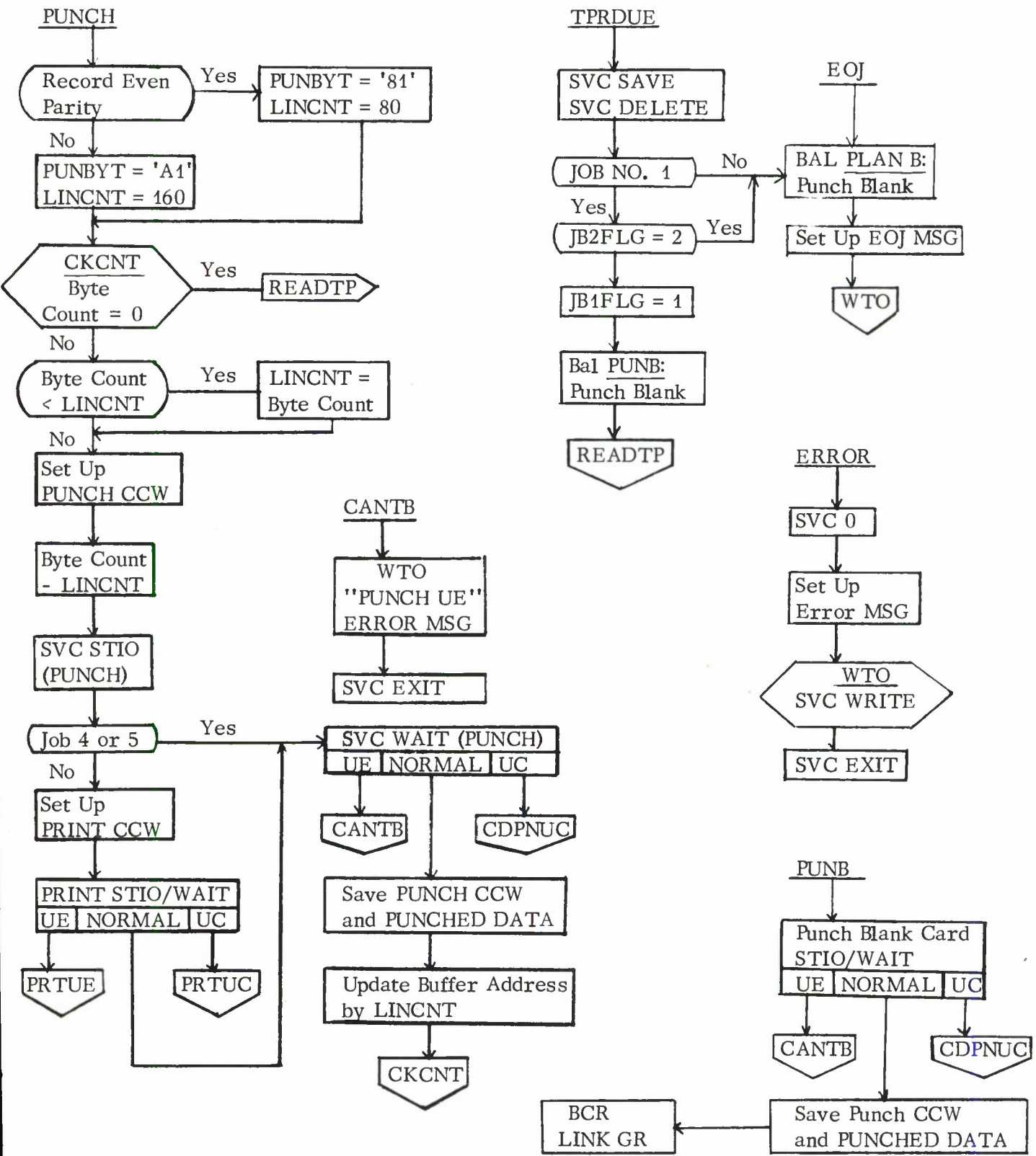
JOB	FILES	RECORDS
1	0	3
2	0	3
3	2	0
4	2	2

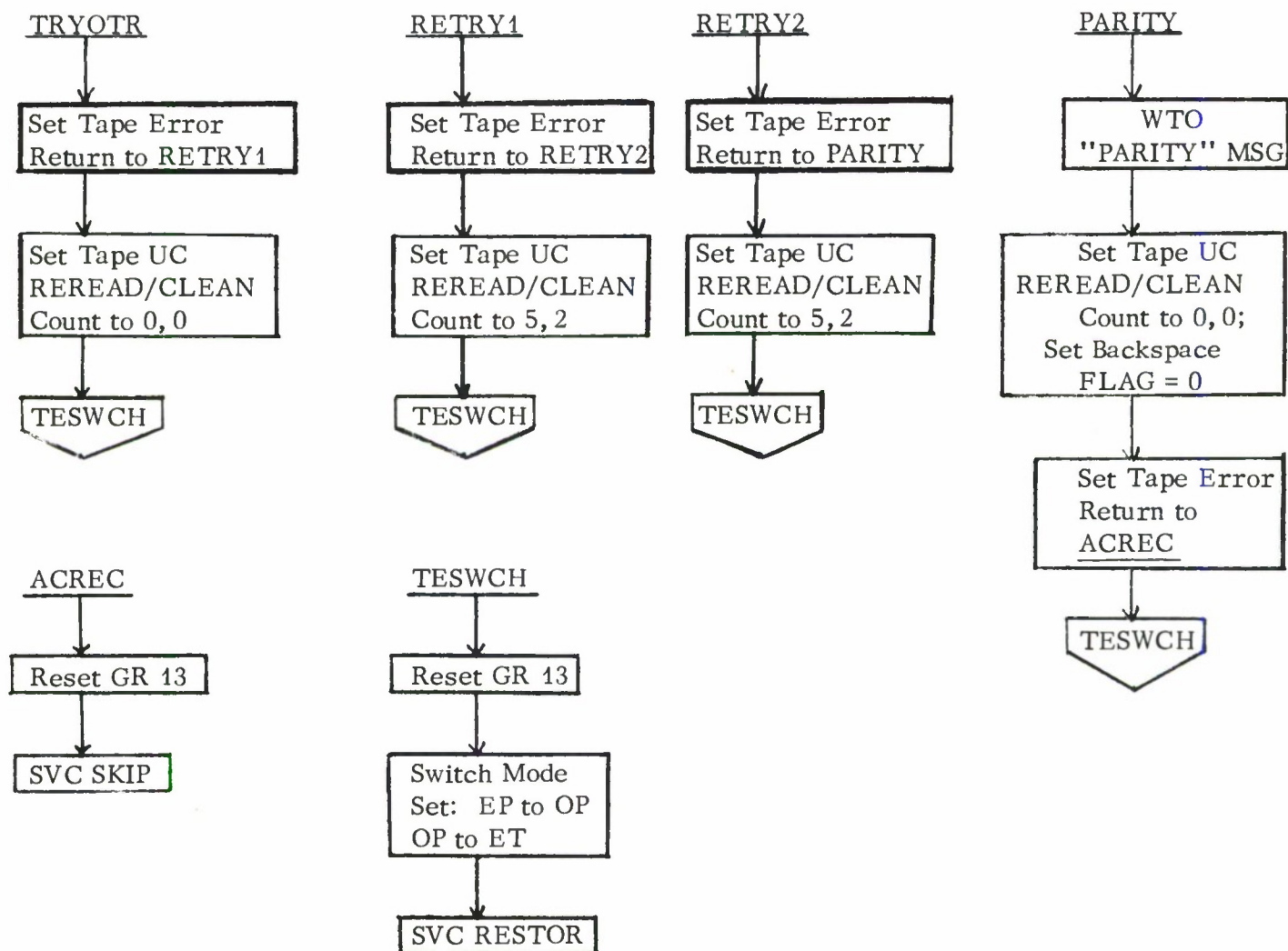
It is essential, therefore, that the tape be positioned at load point upon job initiation.

The following is a flow chart of the SHARE program.









Lincoln Laboratory Multi-Programming Supervisor

18 April 1966

IV. The Job Program

- A. Writing a Job Program
- B. Supervisor Call Summary
- C. Supervisor Calls
- D. Job Program Subroutines (Unit Check Routines)

Lincoln Laboratory Multi-Programming Supervisor

15 July 1966

IV-A. Writing a Job Program

1. The Job Header
2. Entry to a Job Program
3. The Job List
4. Use of Job Sense Switches

1. The Job Header

Each job program is prefixed by a job header. The job header specifies the location in the job program of the first instruction to be executed, the number of devices and buffers which the job requires, the device name of each device, and the size of each required buffer. The format of the job header is:

```
* Location of First Job Instruction *
*   NJBDVU       *   NJBBFU       *
* Names of Required Devices       *
*       (4 Bytes)                 *
*       ...                       *
*       ...                       *
*       ...                       *
* Sizes of Required Buffers       *
*       (4 Bytes)                 *
*       ...                       *
*       ...                       *
*       ...                       *
```

NJBDVU = Number of Devices Used

NJBBFU = Number of Buffers Used

An illustrative 360 coding sequence of a job using two devices and three buffers is shown below.

```
JOB   START      0
      DC          A(BEGADR)
      DC          H'2'
      DC          H'3'
      DC          CL4'PTR'
      DC          CL4'7TP'
      DC          F'128'
      DC          F'2048'
      DC          F'2048'
      .
      .
      .
BEGADR EQU          *
```

The number of required devices are specified in the field NJBDVU and the number of required buffers are specified in the field NJBBFU. The device names for each required device must be given in the full words following the word specifying the number of devices. The size of each required buffer must be given in the full words following the device names.

Device names, specifying the device requirement, are four characters, left justified, with trailing blanks. Device names are:

RDR	-	2540 Card Reader
PCH	-	2540 Card Punch
PTR	-	1403 Printer
7TP	-	7-Track Tape Unit
9TP	-	9-Track Tape Unit
TP	-	Either 7-Track or 9-Track Tape Unit
DISK	-	2311 Disk Unit
PDA	-	2701 Parallel Adapter
PLOT	-	Calcomp Plotter
SCOP	-	2250 Display Scope
DISP	-	2260 Display Terminals
TYPE	-	2741 Typewriter Terminal
TTY	-	Teletype Terminal
CONS	-	1050 Console Typewriter.

The order in which the device names are specified determines a logical device number (LDN) for each device, where the first device specified is logical device one. When a job program issues a supervisor call, the device to which the call is associated is indicated by the logical device number. In this way a job program can be written independently of the physical address of a device.

2. Entry to a Job Program

When a job program is successfully initiated from the console typewriter, control is passed to the first instruction as specified in the Job Header. Three locations in the Job Table associated with the job are indicated in General Registers 0, 1 and 2. General Register 0 contains the address of the pseudo Sense Switches, General Register 1 contains a pointer to the list of buffer addresses, and General Register 2 contains a pointer to the list of input parameters. If an input parameter is alphanumeric, it is right justified with leading blanks. If an input parameter is a decimal integer, it is converted to a four byte signed binary number. The list of input parameters is terminated with a fence of FFFFFFFF. If there are no input parameters, the first word of the parameter list will be the fence.

3. The Job List

For each job which is to be run under the supervisor, there must be an entry in the Job List. The Job List consists of a collection of Job List Entries. The Job List Entry indicates whether the job is re-entrant and gives the location of the job. Each Job List Entry is assembled as a separate subprogram, and contains the Entry name of the job program as an Extern in the Job List Entry Subprogram. The format of a Job List Entry is:

*		(Job Name)	*
*		(8 characters)	*
*	Version	* Flag * Task number	*
*		* Job program address	*

An illustrative 360 coding sequence for assembling a Job List Entry is shown below:

AUTHOR	START	0	
	EXTRN	JOB	
	DC	CL8' JOBNAME'	job name right justified
	DC	C' 1'	version number
	DC	X' 80'	re-entrant job flag on
	DC	H' 0'	space for task number
	DC	A(JOB)	location of job header
	END		

The byte labeled Version can be used to indicate that a modification has been made to the job program. In the illustrative coding above, Version 1 is indicated. The Flag byte specifies whether the job is re-entrant. Job programs may be written as re-entrant, whereby a single copy of a job program can be active for more than one task. If a job is not re-entrant, there must be a separate copy of the job program and job list entry subprogram for each possible concurrent activation of the job. If a job is re-entrant, the left most bit of the flag byte is set to 1.

Associated with every active job is a task number which is used to identify the particular activation of the job throughout the system. If a job is not re-entrant, it can be active for only a single task. To indicate that a nonre-entrant job is active, the supervisor inserts the task number in the indicated field of the Job List Entry.

Each subprogram which makes up the supervisor deck must have a unique name. Consequently, each Job List Entry subprogram must also have a unique name. It is suggested that the Job List Entry subprogram be given the name of the programmer of the job program. It is up to the job programmer to use a distinct name for each of his Jobs and Job List Entry subprograms. The name of each Job List Entry subprogram might be formed from the author's name and a sequence number.

4. Use of Job Sense Switches

All job programs should be written so as to place flag bits used for control into the sense switch area. Not only does this favor re-entrancy, but it also permits dynamic control of job parameters. For example, the PRINT job PC, PCE, SP 1, and SP 3 options could vary between files if they were represented by sense switch settings. The operator could thus vary the mode of printing without terminating and re-initiating the task simply by invoking the SWITCHES job.

It is further recommended that all job programs include the following standard sense switch meanings and use them to permit a graceful interruption of a task:

<u>Sense Switch Bit</u>	<u>Function</u>
0	Terminate immediately
1	Terminate at end of record
2	Terminate at end of file
3-6	Reserved
7	Pause
8-31	As desired .

The terminations accomplished by bits 1 and 2 should be "clean;" that is, they should allow the job to be re-initiated without producing any different results than if the task had been allowed to continue. The pause function of bit 7 provides a means for suspending the task. It should be implemented by coding similar to the following:

PAUSE	SVC	DORMNT	
	TM	SWITCH, 1	
	BC	1, PAUSE	
CONTIN	EQU	*	continue after switch is 0.

The use of switches for a job should be described in the operating instructions.

5. Buffer Sizes

The buffers available in the current LLMPs systems (1 March 1967) are as follows:

A. Standard Systems

- 1) 6 buffers of 96 bytes
- 2) 6 buffers of 200 bytes
- 3) 3 buffers of 512 bytes
- 4) 4 buffers of 1024 bytes
- 5) 8 buffers of 2048 bytes
- 6) 3 buffers of 4096 bytes

B. Large Buffer Systems

- 1) 6 buffers of 96 bytes
- 2) 9 buffers of 200 bytes
- 3) 3 buffers of 512 bytes
- 4) 3 buffers of 1024 bytes
- 5) 7 buffers of 2048 bytes
- 6) 2 buffers of 27656 bytes

C. 360/67 Systems

- 1) 16 buffers of 256 bytes
- 2) 16 buffers of 512 bytes
- 3) 16 buffers of 1024 bytes
- 4) 16 buffers of 2048 bytes
- 5) 4 buffers of 32768 bytes

Lincoln Laboratory Multi-Programming Supervisor

11 April 1966

IV-B. Supervisor Call Summary

Attached are two forms of a summary sheet for all present job supervisor calls. One form gives the supervisor calls and their calling sequences in tabular form; the other form specifies the call sequences by illustrative 360 calling sequences.

Symbols used in the Supervisor Call Summary

CTLST	-	dump control list address
LDN	-	logical device number for the unit
CAW	-	channel address of a command
RTNLST	-	location of returns list
COUNT	-	residual count
LMESAG	-	number of bytes in message
MESSAGE	-	location of text to be written on the on-line typewriter
BUFFER	-	location of read buffer
CHARCT	-	number of characters read
SNSBUF	-	location of sense buffer
DEVID	-	device identification
SVQMTY	-	saved I/O queue empty
CANCEL	-	read terminated by a CANCEL
SNSOK	-	sense bytes stored
SNSBZY	-	sense command rejected because of busy device
SNSUC	-	UC on sense command .

Supervisor Call Summary

SVC	NAME	FUNCTION	Return Value			Special Return Point	CC Affected
			GR 0	GR 1	GR 2		
0	JOBDMF	Job Dump					
1	DMPLST	Control List Dump	CTLST				
2	STIO	Add to I/O Queue	LDN	CAW	RTNLST		
3	WAIT	Wait on I/O	LDN			*	†
4	DORMNT	Release CPU					
5	RESCNT	Get Residual Count	LDN				
6	END	Terminate Task				**	
7	WRITE	Give Message to Operator	LMESAG	MESSAGE			
8	SAVE	Clear Active I/O Queue	LDN				
9	RESTOR	Restore Saved I/O Queue	LDN			***	
10	DELETE	Delete Top Saved Entry	LDN				††
11	READ	Read Operator Input		BUFFER	CHARCT		†††
13	SKIP	Accept I/O Error	LDN			***	
16	SPWAIT	Wait for Stray Device End	LDN		RTNLST	*	
18	SAVSKP	SAVE followed by SKIP	LDN			***	
19	SNSADR	Start Automatic Sensing	LDN	SNSBUF			
20	GETID	Get Device Identification	LDN		DEVID		

* On a UE or UC condition the return is as specified in the returns list.

** No return to the job program.

*** Return is to the SVC WAIT associated with the top entry on the saved I/O queue.

† CC = 0 for non-immediate operations; 3 for immediate operations; and 1 for initial rejection.

†† CC = 0 if saved I/O Queue is empty; 1, if not.

††† CC = 0 if operator pressed cancel; 1, if not.

SVC Calling Sequences

The calling sequences are specified by illustrative 360 coding sequences where the following definitions are used:

CTLST	-	location of dump control list
LDN	-	logical device number for the unit
CAW	-	location containing the channel address of a command
RTNLST	-	location of returns list
COUNT	-	location to store residual count
LMESAG	-	number of bytes in message
MESSAGE	-	location of text to be written on the on-line typewriter
BUFFER	-	location of read buffer
CHARCT	-	location to store number of characters read
SNSBUF	-	location of sense buffer
DEVID	-	location to store device identification .

Instructions transferred to after testing condition codes:

SVQMTY	-	saved I/O queue empty
CANCEL	-	read terminated by a CANCEL
SNSOK	-	sense bytes stored
SNSBZY	-	sense command rejected because of busy device
SNSUC	-	UC on sense command .

SVC Calling Sequences

0) SVC	JOBDMF	11) LA	1, BUFFER
1) LA	0, CTLST	SVC	READ
SVC	DMPLST	ST	0, CHARCT
		BC	CANCEL
2) LA	0, LDN	13) LA	0, LDN
LA	1, CAW	SVC	SKIP
LA	2, RTNLST	(see note**)	
SVC	STIO		
3) LA	0, LDN	16) LA	0, LDN
SVC	WAIT	LA	2, RTNLST
(see note*)		SVC	SPWAIT
		(see note*)	
4) SVC	DORMNT	18) LA	0, LDN
5) LA	0, LDN	SVC	SAVSKP
SVC	RESCNT	(see note**)	
ST	0, COUNT		
6) SVC	END	19) LA	0, LDN
(no return)		LA	1, SNSBUF
		SVC	SNSADR
		:	
7) LA	0, LMESAG	(after UC)	
LA	1, MESSAGE		
SVC	WRITE	CLI	SNSBUF, 1
8) LA	0, LDN	BC	8, SNSOK
SVC	SAVE	BC	4, SNSBZY
		BC	2, SNSUC
9) LA	0, LDN	20) LA	0, LDN
SVC	RESTOR	SVC	GETID
(see note**)		ST	0, DEVID
10) LA	0, LDN		
SVC	DELETE		
BC	8, SVQMTY		

* on a UE or UC condition after a STIO, the return is as specified in the returns list of the associated STIO.

** return is to the SVC WAIT associated with the top entry on the saved I/O queue.

Lincoln Laboratory Multi-Programming Supervisor

13 April 1966

IV-C. Supervisor Calls

0	JOBDMF
1	DMPLST
2	STIO
3	WAIT
4	DORMNT
5	RESCNT
6	END
7	WRITE
8	SAVE
9	RESTOR
10	DELETE
11	READ
13	DKIP
16	SPWAIT
18	SAVSKP
19	SNSADR
20	GETID

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Job Dump

1 January 1967

Name: JOBDMP

Supervisor Call: 0

Function:

To get a dynamic dump, for debugging purposes, of an active job program together with its buffers and the supervisor tables which contain status information of the job.

Calling Sequence: None

Return Value: None. The dump is output on printer F on the model 40, and the single printer on the model 65.

Error Exit:

Up to 10 dumps may be stacked. If the JOBDMP SVC is called while 10 dumps are stacked, a supervisor error condition occurs.

Description:

The dump produced includes the JOBLIST, JOBTABLES, DEVICE LIST, DEVICE TABLES, and ENTRY POOL, the CPUQ and the first 4096 bytes of the job program as well as each buffer.

The first page of the dump gives the low core system registers, first 128 (hex) bytes, and is relevant, primarily, to the supervisor. It includes: the first 24 IPL locations: interrupt old and new PSW's: the CSW, CAW, and TIMER registers; and the 2 unused words of low core before and after the TIMER register. The register after the TIMER register contains the job table location of the job being dumped. In addition, a message is provided indicating the job name associated with this dump and the error code (RQST).

The second, third, fourth, and fifth pages give the complete JOBLIST, the JOB TABLE, the DEVICE LIST, and all DEVICE TABLES as used by the supervisor. Table formats can be obtained from the supervisor listings. Of particular interest, the Job Table contains the Device Table Locations of the devices used, the initial input parameters, and the jobs general registers as they were at the time of the supervisor call.

Next comes the entry pool containing I/O entries and CPUQ entries, followed by the CPUQ. The format of the entries can be obtained from the supervisor listings.

The first CPUQ entry address can be obtained from the CPUQ. The first word of the CPUQ Entry contains the second half of the task's current PSW. The address of I/O queue entries can be obtained from the device table of a particular device. An I/O Entry contains a CAW and an I/O returns list location.

The first 4096 bytes of the job program are dumped beginning with the job header in half-word, with mnemonic format. The storage areas used by the job program are then dumped in their entirety.

Note:

Whenever the supervisor recognizes a job error, an automatic job dump is provided and the job is terminated. The type of error can be determined from the error code. The following error codes can occur:

SVC	-	illegal SVC execution
PGNT	-	program interruption
CSW	-	channel program error
ILDN	-	illegal logical device number .

When a program interruption error occurs, the general registers at the time of the interruption are given before the dump of the job program. The job table contains the contents of the general registers as of the last previous SVC or I/O interruption and the CPUQ holds the second-half of the PSW as of this last previous interruption.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Dump List

1 January 1967

Name: DMPLST

Supervisor Call: 1

Function:

To obtain a control list dump.

Calling Sequence:

GR 0 - Pointer to a single dump control list.

Return Value:

None. The dump is output on Printer F.

Error Exit:

If the address of the control list exceeds the addressable memory of the machine, the job is terminated with the error code SVC. The supervisor permits 10 dump requests to be stacked, and if a dump request is made while 10 requests are stacked, a supervisor error condition occurs.

An improper control list causes an error comment to be printed and the request to be skipped.

Description:

The control list specifies the type of dump desired. It consists of 8 control bytes (double word aligned) followed by an optional 8-character label. (Refer to pp. 28-34, Form C28-6505, BPS Basic Utilities.) Each control list specifies the following:

- a) whether or not there is a dump identification label,
- b) whether the End Flag field specifies an end address +1 location or a count of bytes,
- c) the format code of the dump,
- d) the address of the first byte of the area to be listed,
- e) the length in bytes of each item (where not implied by the format),
- f) the size of the area as defined in the End/Count field,
- g) the 8-byte label.

The Control List Entry format is shown below:

0 - 3	4 - 7	8 - 31	32 - 39	40 - 63	64 - 127
Label End Flag	Format Code	First Address	Length	Last Address or Count	Label

In the coding of a Control List Entry the Label Flag and End Flag are specified together in one hexadecimal character as follows:

Label/End Flag:	
f =	$\left\{ \begin{array}{ll} 0 & - \text{No Label/Count} \\ 4 & - \text{No Label/Address} \\ 8 & - \text{Label/Count} \\ C & - \text{Label/Address} \end{array} \right.$

Format Codes:

c =	$\left\{ \begin{array}{ll} 0 & - \text{Hexadecimal with length specified in Length field.} \\ 1 & - \text{Character with length specified in Length field.} \\ 2 & - \text{Packed decimal with length specified in Length field.} \\ 3 & - \text{Zoned decimal character with length specified in Length field.} \\ 4 & - \text{Hexadecimal half-word with mnemonics.} \\ 5 & - \text{Hexadecimal full-words without mnemonics. Length of each item is four bytes.} \\ 6 & - \text{Short-precision floating-point decimal.} \\ 7 & - \text{Long-precision floating-point decimal.} \\ 8 & - \text{Same as 4.} \\ 9 & - \text{Half-word fixed-point decimal.} \\ A & - \text{Full-word fixed-point decimal.} \end{array} \right.$
-----	--

Control List Entry

DC	X'fc	f = Label/End Flag, c = Format Code
DC	AL3(aaaa)	First address to be dumped
DC	AL1(nnn)	Length of bytes to be grouped in format (0 to 255)
DC	AL3(bbbb)	Last address +1 or count of items
DC	CL8'label'	Dump label.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Start I/O

1 January 1967

Name: STIO

Supervisor Call: 2

Function:

Adds a command to the I/O queue for a specified device to initiate an I/O operation.

Calling Sequence:

- GR 0 - Logical device number (1 - 5).
- GR 1 - Address of an I/O channel command.
- GR 2 - Location of returns list.

The returns list consists of 3 full words aligned at a word boundary. The first word is storage for the normal return after an SVC WAIT, the second word contains the return address for a unit exception condition, and the third word contains the return address for a unit check condition.

Return Value:

None. Return is to the instruction following the SVC.

Error Exit:

The job is terminated whenever there is an error in a calling sequence parameter. An illegal logical device number causes the job to terminate with the code ILDN. If the address of a CCW or the location of returns list exceeds addressable memory, the job terminates with the code SVC. An error in a channel command will cause the job to terminate with the code CSW.

Description:

An I/O entry is added to the I/O queue for the device unit indicated. An available entry is obtained from the entry pool and is linked to the active I/O queue. The last entry address is set in the device table and, if the active queue was empty, the first and active entry address is set to point to the new I/O entry. The supervisor then attempts to start the channel command program.

Note:

Command and data chaining are permitted in a channel program. The incorrect length interrupt is ignored and not passed back to the job program. A normal return, indicated by a device end from the device, or a unit check or unit exception condition, results in an appropriate return to the job. Attention and program-controlled interrupts cause immediate job termination with code 'CSW'.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Wait

1 January 1967

Name: WAIT

Supervisor Call: 3

Function:

Provides interlock to wait for a device to signal the completion of an I/O operation.

Calling Sequence:

GR 0 - Logical device number.

Return Value:

CC = 0 - non-immediate operation
CC = 1 - initial rejection of the command
CC = 2 - unused
CC = 3 - immediate operation.

Each SVC WAIT is associated with an SVC STIO which added an I/O request to the active I/O queue in the Device Table of the device indicated by the logical device number. When the I/O command initiated by the SVC STIO results in a unit exception (UE) or unit check (UC) condition, the return is to the appropriate address indicated in the return list specified by the associated STIO. Otherwise, the return is to the instruction following the SVC WAIT.

Error Exit:

The job is terminated with an error code ILDN if the logical device number is incorrect. An SVC WAIT not preceded by an associated SVC STIO causes the job to terminate with a code SVC.

Description:

SVC WAIT permits processing to continue after the completion of an I/O operation. The instruction after the SVC WAIT is taken to be the normal return and its address is inserted in the returns list specified by the STIO associated with the SVC WAIT. The abnormal returns for unit exception (UE) and unit check (UC) are specified in the returns list. For each STIO supervisor call, there must be a corresponding WAIT supervisor call, and visa versa.

Upon return from the SVC WAIT, the condition code will have been set to indicate if the command was initially rejected (CC = 1) or, if accepted, whether or not it was an immediate operation. A condition code of 1 can only occur with an abnormal return.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Dormant

1 January 1967

Name: DORMNT

Supervisor Call: 4

Function: Places the current job at the bottom of the CPUQ.

Calling Sequence: None

Return Value: None

Error Exit: None

Description:

The current job, represented by the given entry at the top of the CPUQ, is placed on the bottom of the CPUQ. This call permits other jobs to have access to the central processor.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Residual Count

1 January 1967

Name: RESCNT

Supervisor Call: 5

Function: Obtains the residual count of a command after data transmission.

Calling Sequence:

GR 0 - Logical device number

Return Value:

GR 0 - Residual Count

Error Exit:

An illegal logical device number causes the job to terminate with a code of ILDN.

Description:

On the receipt of a Channel End interrupt, the residual count from a command is stored in a temporary location (RDVCNT) in the device table of the interrupting device.

On a normal device end, the temporary count is stored in the CAW part of the corresponding I/O queue entry. When the normal return is added to the CPU queue, the count is transferred from the queue entry back to the device table, stored as the residual count (in location RDVCNT), and may be obtained from a job program as the residual count from the previous operations.

On a unit check (UC) or unit exception (UE) condition the temporary residual count is stored as the residual count (in location RDVCNT) for the abnormal condition. The residual count may then be obtained from a job program by a call to RESCNT.

This procedure of storing the residual count in temporary locations is necessary since more than one operation may be added to the I/O queue of one device, and, for each operation, a residual count must be saved. The residual count which is obtained with the supervisor call is associated with the last I/O operation completed on the device. The residual count can be used to determine the number of bytes that were actually transmitted or received in an operation.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC End

1 January 1967

Name: END

Supervisor Call: 6

Function: Terminates a job.

Calling Sequence: None

Return Value: None

Error Exit: None

Description:

The job is terminated and all its devices are freed. All jobs should call END after its function has been completed.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Write

1 January 1967

Name: WRITE

Supervisor Call: 7

Function: Writes a message on the console typewriter.

Calling Sequence:

- GR 0 - Number of characters in message.
- GR 1 - Location of first character in message.

Return Value: None. Return is immediate.

Error Exit:

The job terminates with the code SVC if the address of the last character of the message is beyond the bounds of core memory.

Description:

The message specified in the calling sequence is printed on the console typewriter preceded by the job number and the job name. If the number of characters in the message is greater than 100 characters, the last 97 characters are printed followed by the characters 'xxx' to indicate that the line has been truncated.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Save

1 January 1967

Name: SAVE

Supervisor Call: 8

Function: Transfers Active I/O queue to top of Saved I/O queue.

Calling Sequence:

GR 0 - Logical device number

Return Value: None

Error Exit:

The job is terminated with the code ILDN if the logical device number is illegal, and with code SVC if the SVC SAVE is issued other than in the processing of a unit exception of unit check condition after an SVC WAIT.

Description:

SVC SAVE stacks the Active I/O queue onto the Saved I/O queue, leaving the Active I/O queue empty. With the Active I/O queue cleared a new I/O operation may be issued using the SVC STIO, or the SVC SPWAIT may be issued to wait for a stray device end signal. The I/O entry (or entries) on the Saved I/O queue must eventually be returned to the Active I/O queue or deleted.

An entry is added to the Active I/O queue on each SVC STIO and is removed from the Active I/O queue on the successful (normal) completion of the operation. When an abnormal condition occurs, the abnormal return (UE or UC) is taken but the I/O entry is not removed from the queue. Subsequent action must make use of the SVC SAVE followed by at least one of the following:

- a) SVC SKIP
- b) SVC DELETE
- c) SVC RESTOR

An SVC SAVE should not be issued while a device is busy. That is, it should be issued only after an abnormal return from an SVC WAIT.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Restore

1 January 1967

Name: RESTOR

Supervisor Call: 9

Function: Transfers Saved I/O queue to bottom of Active I/O queue.

Calling Sequence:

GR 0 - Logical device number.

Return Value:

None. Return is to the SVC WAIT associated with the top entry of the I/O queue.

Error Exist:

The job is terminated with code ILDN if the logical device number is illegal and with code SVC if the resulting Active queue is empty.

Description:

SVC RESTOR chains the Saved I/O queue to the bottom of the Active I/O queue, leaving the Saved I/O queue empty. The top entry on the I/O queue is then used to activate the device. The job is then put in WAIT status at the SVC WAIT associated with the top entry on the I/O queue waiting for the completion of the I/O operation.

This supervisor call may be used when it is desired to redo an I/O operation after an abnormal condition. It should not be called if the resulting Active I/O queue would be empty.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Delete

1 January 1967

Name: DELETE

Supervisor Call: 10

Function:

Removes the top entry on the Saved I/O queue.

Calling Sequence:

GR 0 - Logical device number.

Return Value:

CC = 0 Saved I/O queue empty.

CC = 1 Saved I/O queue not empty.

Error Exit:

The job is terminated with the code ILDN if the logical device number is illegal.

Description:

SVC DELETE removes the top entry on the Saved I/O queue and sets the condition code to zero if the Saved queue is empty, and to one otherwise. If it is desired to delete the complete Saved queue, this Supervisor Call must be issued repeatedly until the queue is empty.

If the queue is already empty, the supervisor call is ignored.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Read

1 January 1967

Name: READ

Supervisor Call: 11

Function: Reads an operator message.

Calling Sequence:

GR 1 - Buffer location for message.

Return Value:

GR 0 - Number of characters read.
CC=0 - Message was terminated by CANCEL.
CC=1 - Message received.

Error Exits:

The job is terminated with code SVC if the buffer location is within 100 locations of the end of core.

Description:

The job number, job name, and three dots (...) are printed on the typewriter. The proceed light is then turned on permitting an operator message to be typed. The message may be terminated by either a carrier return (CR), End of Block (EOB), or CANCEL signal. The message is automatically terminated on receipt of the 100th character. The number of characters read is returned in GR 0. If the message was terminated by a CANCEL signal, the condition code is set to 0, otherwise it is set to 1.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Skip

1 January 1967

Name: SKIP

Supervisor Call: 13

Function: Forces a normal return from an I/O entry on the Saved I/O queue.

Calling Sequence:

GR 0 - Logical device number

Return Value:

None. Return is to the instruction after the SVC WAIT (the normal return) associated with the top entry of the Saved I/O queue.

Error Exit:

The job is terminated with code ILDN if the logical device number is illegal, or with code SVC if the Saved I/O queue is empty.

Description:

The supervisor call SKIP is used to accept an abnormal condition and to return to the normal return associated with the operation. It is to be issued only when there is an entry on the Saved I/O queue. The Saved I/O queue is added to the bottom of the Active I/O queue, clearing the Saved I/O queue, and the top entry in the I/O queue is then removed and its normal return added to the CPU queue.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Special Wait

1 January 1967

Name: SPWAIT

Supervisor Call: 16

Function: To wait for a device end signal for a device.

Calling Sequence:

- GR 0 - Logical device number.
- GR 2 - Location of returns list.

The returns list, like that used with the SVC STIO, consists of 3 full words aligned at a word boundary. The first word is storage for the normal return, the second word contains the return address for a unit exception condition, and the third word contains the return address for a unit check condition.

Return Value:

When a stray device end comes from the specified device, return is as indicated in the returns list. The normal return is to the instruction following the SVC SPWAIT.

Error Exit:

The job is terminated whenever there is an error in a calling sequence parameter or if the SVC SPWAIT is issued with a non-empty I/O queue.

Description:

SVC SPWAIT is used to provide an interlock to wait on a stray device end signal. A stray device end signal is generated whenever the device goes from a not ready to a ready state. Thus, this supervisor call can be used to wait for a not ready device to become ready.

This supervisor call is like the supervisor call SVC STIO for no operation, followed by the SVC WAIT to wait for the device end signalling the completion of the operation, i. e. , the device is ready. No I/O operation can be in progress when this supervisor call is issued otherwise an SVC error exit will be taken.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Save-Skip

1 January 1967

Name: SAVSKP

Supervisor Call: 18

Function: Forces a normal return after an abnormal condition.

Calling Sequence:

GR 0 - Logical device number.

Return Value:

None. Return is to the instruction after the SVC WAIT (the normal return) associated with the top entry of the Active I/O queue.

Error Exit:

The job is terminated with code ILDN if the logical device number is illegal, or with code SVC if the call is issued other than in the processing of a unit exception or unit check condition after an SVC WAIT.

Description:

This supervisor call is a combination of SVC SAVE followed by SVC SKIP. It may be used when it is desired to accept an abnormal condition and take the normal return from an I/O operation.

The Active I/O queue is stacked onto the Saved I/O queue, leaving the Active I/O queue empty, and then the Saved I/O queue is transferred to the Active I/O queue, leaving the Saved I/O queue empty. The top entry in the I/O is then removed and its normal return is added to the CPU queue.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Sense Address

1 January 1967

Name: SNSADR

Supervisor Call: 19

Function: Controls automatic sense on unit check.

Calling Sequence:

GR 0 - Logical device number.

GR 1 - Pointer to sense area; this value is zero
if no automatic sensing is desired.

Return Value:

None. The contents of the general registers and the value of the condition code remain unchanged. The first byte of the sense area is set to zero if GR 1 did not contain a zero.

Error Exit:

The job is terminated if the logical device number is incorrect, in which case the code ILDN is displayed. If the sense area address exceeds the addressable memory of the machine, the job is terminated with the error code SVC.

Description:

The address in GR 1 is stored in the device table and the automatic-sense flag is set in the device table. If a unit check indication is received from this device, the supervisor issues a sense command to the device, using the sense area address. If the sense command is successful, the sense information is stored starting at the second byte of the sense area. The first byte is used as a flag to indicate the results of the sense command, and is set as follows:

- 0 - Sense command was rejected because the device was still busy from a previous operation.
- 1 - The sense command was successfully completed.
- 2 - Unit check was received while issuing the sense command; three attempts were made.

Note that a flag setting of 0 cannot usually occur with tape units; it might occur in operations for which channel end and device end are separate interrupts, and where unit check can appear with channel end.

Up to six bytes of sense information are transferred, the actual number being determined by the device type.

Note that this supervisor call needs to be issued only once in the running of a task if the automatic sense feature is always desired.

Lincoln Laboratory Multi-Programming Supervisor

Supervisor Call Description

SVC Get ID

1 January 1967

Name: GETID

Supervisor Call: 20

Function: Obtains device identification.

Calling Sequence:

GR 0 - Logical device number.

Return Value:

GR 0 - Four character device identification; four hyphens (----) are returned for devices having only one of a type. All other general registers and the value of the condition code remain unchanged.

Error Exit:

The job is terminated with the error code ILDN if the logical device number is incorrect.

Description:

The four character device identification as typed in by the operator is returned in GR 0. It may then be placed in a message to be displayed to the operator to help him identify the unit.

Lincoln Laboratory Multi-Programming Supervisor

5 April 1966

IV-D. Job Program Subroutines (Unit Check Routines)

The system subroutines are for use by jobs operating under the Lincoln Laboratory Multi-Programming Supervisor. These routines are re-entrant and can thus be called by any number of jobs.

The routines described are:

1. Tape Utilities
ERG, WTM, FSR, BSR, FSF, BSF, REW, RVN
2. Card Reader Unit Check
CDRDUC, CDSTUC
3. Tape Unit Check
TPWRUC, TPRDUC
4. Printer Unit Exception and Unit Check
PRTUE, PRTUC
5. Find Job List
FJBLST

Lincoln Laboratory Multi-Programming Supervisor

Subroutine Write-up

Tape Utilities (Re-Entrant)

21 March 1966

General Information:

The subroutines described here provide a means for spacing tape forward or backward, (by records or files) erasing tape in units of inter-record gaps, or writing on tape a number of tape marks. The subroutines are re-entrant and are written to operate on 7- or 9-track tape units.

Subroutine Entries:

To call a tape utility subroutine, the entry point must be defined or an EXTRN in the job program and an address constant must be defined for the entry address. A routine is called by loading the entry point address into general register 15 and issuing a BALR 14,15 instruction. The return address will be stored in general register 14 to permit the routine to return to the next instruction. The tape utility subroutines, their functions, and entry names are indicated below.

<u>Name</u>	<u>Entry</u>	<u>Function</u>
Erase Record Gap	ERG	Erase tape for length of one or more inter-record gaps (3-1/2 ")
Write Tape Mark	WTM	Write one or more tape marks
Forward Space Record	FSR	Forward space a tape one or more records
Backspace Record	BSR	Backspace a tape one or more records
Forward Space File	FSF	Forward space a tape one or more files
Backspace File	BSF	Backspace a tape one or more files
Rewind Tape	REW	Rewind tape to load point
Rewind and Unload	RUN	Rewind and unload tape

General Register Settings on Input:

Table IV. D. 1-1 gives the general register settings required for each subroutine where the general registers applicable for each subroutine are indicated by an x in the Table.

The save areas pointed to in GR 4 and GR 13 must be double word aligned. The sense bytes for the tape unit are stored in the first 5 bytes of the sense save area (GR 4) and the CCW is stored in the second double word of this area.

Revised: 1 April 1967

Table IV. D. 1-1. General Register Settings

<u>General Registers</u>	<u>Subroutines</u>							
	ERG	WTM	FSR	BSR	FSF	BSF	REW	RUN
0 - Logical Device Number	x	x	x	x	x	x	x	x
4 - Pointer to 16 byte save area for sense bytes and CCW	x	x	x	x	x	x	x	x
10 - Density code: 0 for 200 bpi, 1 for 556 bpi, 2 for 800 bpi		x	x	x	x	x		
12 - Number of records or files to be spaced, tape marks to be written, or inter-record gaps to be erased	x	x	x	x	x	x		
13 - Pointer to 64 byte general register save area	x	x	x	x	x	x	x	x
14 - Return from subroutine	x	x	x	x	x	x	x	x
15 - Branch to subroutine	x	x	x	x	x	x	x	x

Default Conditions:

1. If density code is not 0, 1, or 2, then a code of 2 (for 800 bpi) is selected automatically.
2. The operation on a 9-track unit forces a density of 800 bpi regardless of the density setting.
3. If the count in GR 12 is less than or equal to 0, a count of 1 is used.

Function Returns:

The subroutines return with the condition code set as indicated in Table IV. D. 1-2 and the sense bytes from the last executed operation stored in the first five bytes of the sense save area pointed to in GR 4. Upon returning from a subroutine, all GR's except GR 12 will contain their original settings. GR 12 will contain the count remaining on termination of the executed routine. The condition codes applicable for each subroutine are indicated by an x in the Table. On returns with condition code 3 the calling routine can check for file protect status by testing sense byte 2 bit 6 and for a not ready condition during operation by testing sense byte 5 bit 1. The bits are numbered 0-7 (from left to right).

Table IV. D. 1-2. Condition Code Settings

<u>Condition Code</u>	<u>Subroutines</u>							
	ERG	WTM	FSR	BSR	FSF	BSF	REW	RUN
0 Operation Successful	x	x	x	x	x	x	x	x
1 Tape mark encountered (End of File)			x	x				
2 Tape marker encountered (End of Tape)	x	x	x		x			
2 Load point marker encountered				x		x		
3 Equipment check or unit made not ready during operation	x	x	x	x	x	x	x	x
3 Unit in file protect status	x	x						

Lincoln Laboratory Multi-Programming Supervisor

Subroutine Write-up

Reader Unit Check and Punch Unit Check-Unit Exception Routines

1 April 1967

General Information:

The card unit check and punch unit exception routine are described in three sections:

- a) Operating instructions describing the messages produced by the routines and the required operator actions,
- b) Programming instructions describing how to use the routines in writing job programs, and
- c) Program descriptions consisting of a flow chart of the card routines and explanatory notes.

Operating Instructions:

Messages come in two forms, depending on the number of card readers or punches attached to the system. If only one reader or punch is attached, the message appears by itself. If more than one reader or punch is attached, the message is preceded by the words "READER RDRn" or "PUNCH PCHn" to identify which reader or punch is being referred to.

Two types of messages can be written by the routines. The first type, which requires operator action, is always associated with a stopped reader or punch and indicates that the card reader or punch is not ready, that a read or punch check has occurred, or that an invalid card code has been read. The second type, which requires that the operator notify either the IBM customer engineer or the responsible system programmer, always contains the word "ERROR" and causes the task to be terminated.

Descriptions of the messages follow:

I. Operator Action Messages

1. READER (or PUNCH) NOT READY

This message indicates that the ready light on the reader (or punch) is not on. If no error lights are lit, insert cards in the hopper and press "start". If the stacker light is on, clear the full stacker and press "start". If the Chips box light is on, empty or replace the punch chip box, whichever is the case, and press "start". If the Feed Stop light on the reader is lit, clear the card

jam. Any cards that have passed the prestacker station on the reader should then be stacked; all other cards should be placed back in the read hopper, and the "start" key pressed. On the punch, it is impossible to recover from a card jam. Therefore, the task should be terminated and the job restarted from the beginning.

2. READ CHECK. RE-FEED THE CARD.
(Reader only)

This message indicates that the card at the reader pre-stacker station has caused a hole count check. The Read Check light will have been turned on. Stop the reader, perform a non-process run-out, and examine the card for misaligned punches. If the card appears to be all right, place it and all following cards in the read hopper and press "start". If the error persists, notify the customer engineer.

3. INVALID EBCDIC CHARACTER. CORRECT AND RE-FEED THE CARD
(Reader Only)

The message indicates that the card at the pre-stacker station contains an invalid EBCDIC code. The validity check light will have been turned on. Perform a non-process run-out and examine the first card. An invalid code is generated whenever any column contains more than one punch in rows 1 to 7. If the card can be corrected, do so (and notify the owner); if it cannot be corrected, but can be ignored (such as a flip card encountered during PRESTORE), place it in the stacker or discard it. In either case, place the run-out cards in the read hopper and press "start". If the card cannot be corrected or ignored, terminate the task.

II. Permanent Error Messages

There are three permanent error messages, and all three cause the task to be terminated. The status information, which consists of four bytes of hexadecimal information from the unit check routines, is displayed for two of the messages (see Programming Instructions). The meaning of the bytes are as follows:

hf	-	HISFLG
mf	-	MYFLAG
sf	-	SNSFLG
sb	-	SENSE (card reader or punch sense byte).

4. MACHINE ERROR CARD READER (or PUNCH) STATUS hfmf sfsb

If this message is printed on the console typewriter, the customer engineer should be notified.

5. PROGRAM ERROR CARD READER (or PUNCH) STATUS hfmf sfsb

If this message is printed on the console typewriter, the responsible system programmer should be notified.

6. MACHINE ERROR -- PUNCH UNIT EXCEPTION

A punch unit exception is not recognized as a legal status condition since the PFR (Punch Feed, Read) feature is not installed on the Lincoln Laboratory 2540 card units. If this message appears, the customer engineer should be notified.

Programming Instructions:

The punch unit check and unit exception routines only handle unit check and unit exception conditions for the punch, feed, select stacker command. It does not handle reliably unit checks and unit exceptions arising from the use of the PFR write command. The reader unit check routine handles card reader unit checks for two types of commands:

1. Read with no feed (entry point CDRDUC)
2. Feed and select stacker (entry point CDSTUC)

Linkage from the calling program requires placing the proper entry point address (CDSTUC, CDRDUC, CDPNUC or CDPNUE), which has been defined as an EXTRN in the calling program, into the returns list as the unit check or unit exception return address. The calling program must also initialize general register 13 to point to a double-word aligned storage area, DSECT1, depicted in Figure IV.D.2-1. The first word in DSECT1 must point to a second double-word aligned storage area, DSECT2, also shown in Figure 1. DSECT1 is used as a work area by the routines and must be 100 bytes long. DSECT2 contains parameters used by the routines and is 12 bytes long for CDPNUE, CDRDUC or CDSTUC and 16 bytes long for CDPNUC.

If any of the unit check return options are selected, the calling program must place the assembled instruction "BC 15, RTNADR" into the first word of DSECT2. RTNADR is the location in the calling program to which control will be returned with the condition code set to 3 when a data check (validity check from the reader) occurs, and with the condition code set to 0 when an equipment check occurs. This full word in DSECT2 must be reserved even if the calling program does not use the options.

The second word contains four bytes of hexadecimal information. The HISFLG byte contains information to be passed from the calling program to the routine. The MYFLAG byte must be cleared by the calling program, and is used by the unit check routine. The SNSFLG and SENSE bytes are used by the supervisor to provide sense information to the unit check routine. The fourth word in DSECT2 is the address of the CCW to be used by the CDPNUC routine in repunching a card upon sensing a hole count error. When a hole count error occurs, it refers not to the card just punched

GR13 contains the address DSECT1

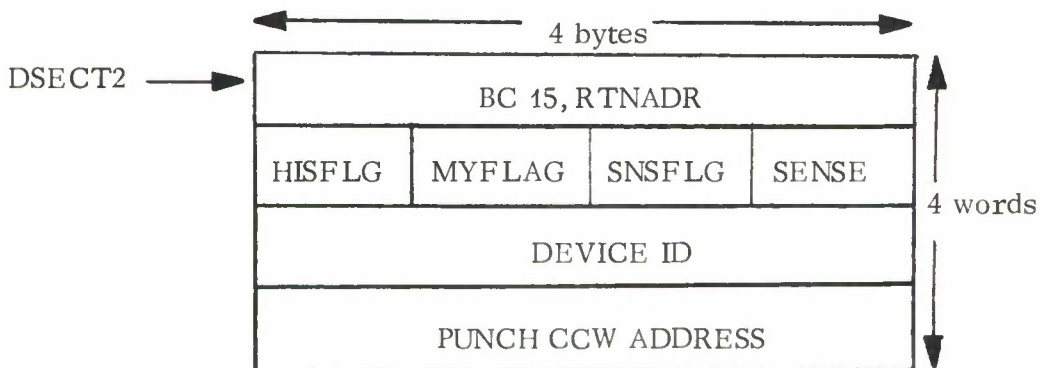
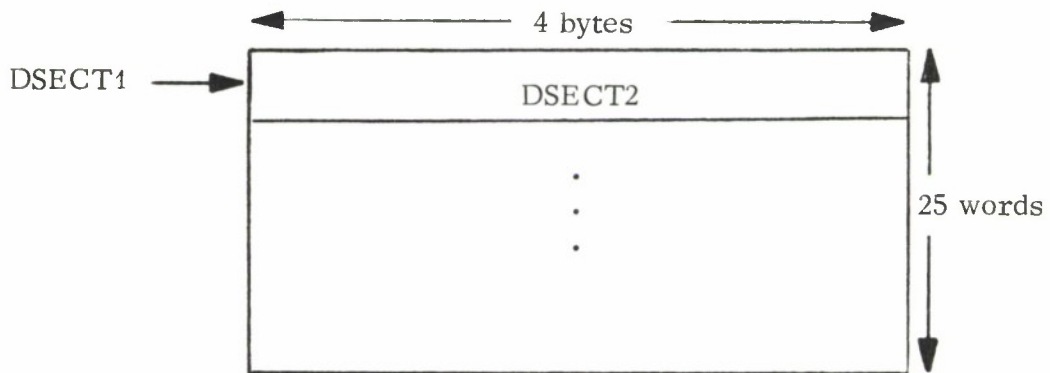


Figure IV.D.2-1. DSECTs used in unit check/unit exception routines.

but to the card punched from the previous write command. Thus, after every SVC WAIT associated with each punch STIO the programmer must save the CCW just used and the data just punched. The address of the saved CCW is placed in the fourth word of DSECT2.

Note that when a hole count error is detected, the card punch unit check routine tries to repunch the card in error. If the error still persists after ten retries a machine error message as described above is typed on the console typewriter. This error recovery procedure assumes that the engineering level of the punch is such that the card following the one in error is automatically directed to stacker 1 after a hole count error.

The location of the SNSFLG byte must be passed to the supervisor via an SVC SNSADR to initiate automatic sensing prior to issuing the first SVC STIO for the card reader or punch. SVC GETID must also be issued, and the device ID obtained placed in the third word of DSECT2. The MYFLAG byte must be cleared prior to each SVC WAIT issued for the card reader or punch. The HISFLG byte is used to select the unit check routine options and to indicate the type of command being performed. The contents of the HISFLG byte are not altered by the routines. For the punch unit exception routine, the user need supply on the device ID field (word three) of DSECT2.

The meaning of the bits in the HISFLG byte (numbered 0-7 from left to right) are as follows:

Bit 0 and 1 - Unused. Must be set to zero.

Bit 2 - PN1BIT - This bit must be set when and only when the first punch command is given. It must be cleared for subsequent punch commands.

Bit 3 - DMSBIT - This bit should be set if a return on a validity check is desired. When a validity check occurs, the unit check routine types message 3 and then issues an SVC SPECWT.

Bit 4 - DATBIT - This bit should be set if a return on a reader validity check is desired without message 3 being typed. Condition code 3 is set following the validity check. This bit overrides the DMSBIT.

Bit 5 - USLBIT - This bit must be set whenever an unusual command sequence is to be issued (i. e., when a read command is issued following a read, no-feed command without a stack command between the read commands).

Bit 6 - BINBIT - This bit must be set whenever a column-binary read command (Data Mode 2) is to be issued to the card reader.

Bit 7 - EQPBIT - This bit should be set if a return on equipment check is desired. When the equipment check occurs, the unit check routine will return to RTNADR with the condition code set to 0.

The meaning of the bits in the MYFLAG byte (numbered 0-7 from left to right) are as follows:

Bit 0 - REPUN - This bit is set by the CDPNUC routine if a unit check occurs on the punch command issued by the routine while attempting to repunch a card.

Bit 1 - PUNBIT - This bit is set when the user enters the card unit check routine via the CDPNUC entry point.

Bit 2 - UCDBIT - This bit is set by the CDSTUC routine when a unit check occurs on the dummy read, no-feed command issued by the routine. (See Section Program Description, 1b, on page IV.D.2-10.)

Bit 3 - DUMBIT or PASS1 - This bit is set by the CDSTUC routine before issuing a dummy read, no-feed command. Or, it may be set by the CDPNUC routine when the maximum repunch count is initialized on a hole count error indication.

Bit 4 - BUSBIT - This bit is set by the card unit check routine when a bus-out check has occurred.

Bit 5 - INIBIT - This bit is set by the card unit check routine when a command was rejected on initial selection.

Bit 6 - REDBIT - This bit is set when the card unit check routine was entered via the CDRDUC entry point.

Bit 7 - STKBIT - This bit is set when the card unit check routine was entered via the CDSTUC entry point.

SENFLG is a code which indicates whether or not the LLMPS supervisor was successful in obtaining sense information on automatic sensing. The codes and their meanings are:

00 - The sense command on automatic sensing was rejected because the device was still busy from a previous operation. For the reader or punch, this code indicates machine error.

01 - The sense command was successfully completed.

02 - Unit check was received while issuing the sense command; three attempts were made. Persistent unit check condition on sensing indicates machine error.

Sample Programs:

An example of the use of the Card Reader unit check routine in a program which reads cards in binary mode, determines whether the card should be read again in BCD mode, and then stacks the cards as shown below:

EXAMPL1	START	0	
	EXTRN	CDRDUC	
	EXTRN	CDSTUC	
	
	LA	0, LDN	Put logical device number of the reader in GR0
	LA	1, SNSFLG	Set up automatic sensing
	SVC	SNSADR	
	SVC	GETID	Get the device ID
	ST	0, DEVID	
	LS	13, DSECT1	
	
BINCRD	LM	0,2, RDBIN	Read card in binary mode
	SVC	STIO	
	MVI	HISFLG,2	Set BINBIT
	MVI	MYFLAG,0	Clear MYFLAG
	SVC	WAIT	
	CLI	BUFF+1,5	Test for 7 and 9 punch in column 1
	BC	7,BCDCRD	
	
BCDCRD	LM	0,2, RDBCD	Read card in BCD mode
	SVC	STIO	
	MVI	MYFLAG,0	Clear MYFLAG
	MVI	HISFLG,20	Set DMSBIT and USLBIT
	SVC	WAIT	
	
STACK	LM	0,2, STAKE	
	SVC	STIO	
	SC	HISFLG(2), HISFLG	Clear HISFLG and MYFLAG
	SVC	WAIT	
	
RTNADR	BC	14, EQPERR	Special processing for validity check
	LA	0,1	
	SVC	DELETE	
	
RDBIN	DC	F'1'	
	DC	A(DRDCCW)	
	DC	A(RDRTN)	

RDBCD	DC	F'1'
	DC	A(BRDCCW)
	DC	A(RDRTN)
STACKE	DC	F'1'
	DC	A(STKCCW)
	DC	A(STKRTN)
RDRTN	DS	F
	DC	A(UEXRTN)
	DC	A(CDRDUC)
STKRTN	DS	F
	DC	A(UEXRTN)
	DC	A(CDSTUC)
DSECT1	DS	0D
	DC	A(DSECT2)
	DS	24F
DSECT2	DS	0D
	BC	15, RTNADR
	BC	15, RTNADR
HISFLG	DS	C
MYFLAG	DS	C
SNSFLG	DS	C
SENSE	DS	C
DEVID	DS	F

An example of the use of the Card Punch unit check and unit exception routines in a program that reads variable unblocked tape records and punches EBCDIC cards is shown below:

EXAMPL2	START	0	
	EXTRN	CDPNUC	
	EXTRN	CDPNUE	
	
	LA	0, LDN	Put logical device number of punch in GR0
	LA	1, SNSFLG	Set up automatic sensing
	SVC	SNSADR	
	SVC	GETID	Get DEVID ID
	ST	0, DEVID	
	LA	13, DSECT1	
	MVI	HISFLG, X'20'	Set flag for 1st punch command
READTP	Read record,
	get physical record length,
	and store in GR6

PUNCH	ST	6, PUNCCW+4	Setup punch CCW
	LM	0, 2, PUNLST	Punch EBCDIC card
	SVC	STIO	
	MVI	MYFLAG, 0	Clear MYFLAG
	SVC	WAIT	
	MVI	HISFLG, 0	Clear initial X'20' in HISFLG
	ST	6, SAVCCW+4	Construct SAVCCW
	BCTR	6, 0	
	EX	6, SAVDAT	Move punched data to save area
	BC	15, READTP	
SAVDAT	MVC	SAVPUN(0), BUFFER	

PUNLST	DC	F'1'
	DC	A(PUNCCW)
	DC	A(PUNRET)
PUNRET	DS	F
	DC	A(CDPNUE)
	DC	A(CDPNUC)
PUNCCW	CCW	X'81', BUFFER, 0, 1
SAVCCW	CCW	X'81', SAVPUN, 0, 1
DSECT1	DC	A(DSECT2)
	DS	24F
	DS	0D
DSECT2	BC	15, RTNADR
HISFLG	DS	C
MYFLAG	DS	C
SNSFLG	DS	C
SENSE	DS	C
DEVID	DS	F
PCHCCW	DC	A(SAVCCW)
SAVPUN	DS	80C

Program Description:

The structure of the card unit check and punch unit exception routines is straightforward and should be easily followed from the flow diagram given in Figure 2. The following notes help to explain the organization of the routines.

1. Separate entries are maintained for the unit check return resulting from a Read, No-Feed command and from a Feed, Select Stacker command for two reasons:

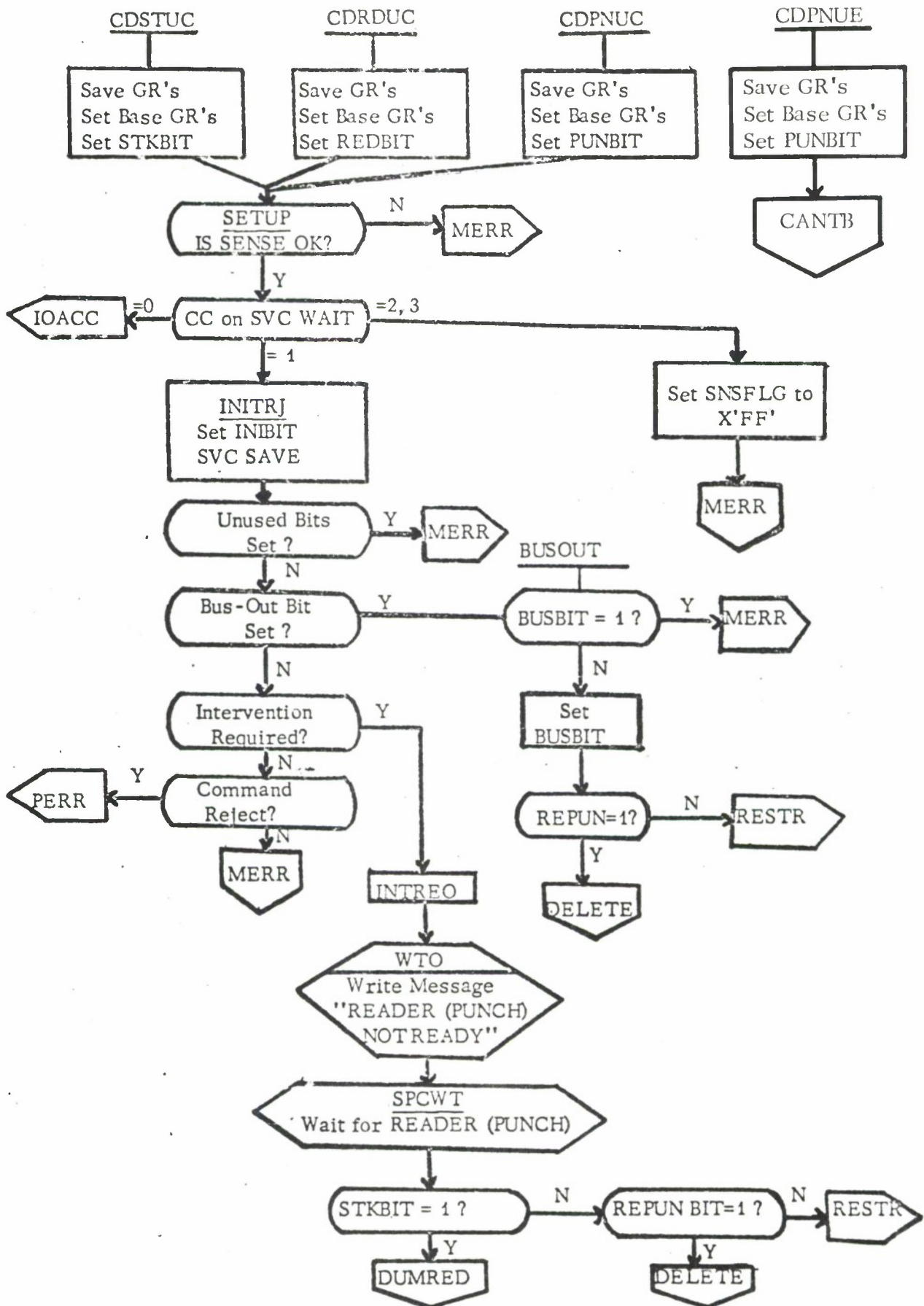
- a. Unit checks for a Feed, Select Stacker command, if they occur, can occur only during the initial selection sequence; and indication of unit check for such a command at other than initial selection is treated as a machine error (see IOACC).
- b. A Feed, Select Stacker command is invalid as the first command following initial run-in on the card reader; thus, a dummy Read, No-Feed command must be executed if the possibility of executing the command after initial run-in exists (see DUMRED).

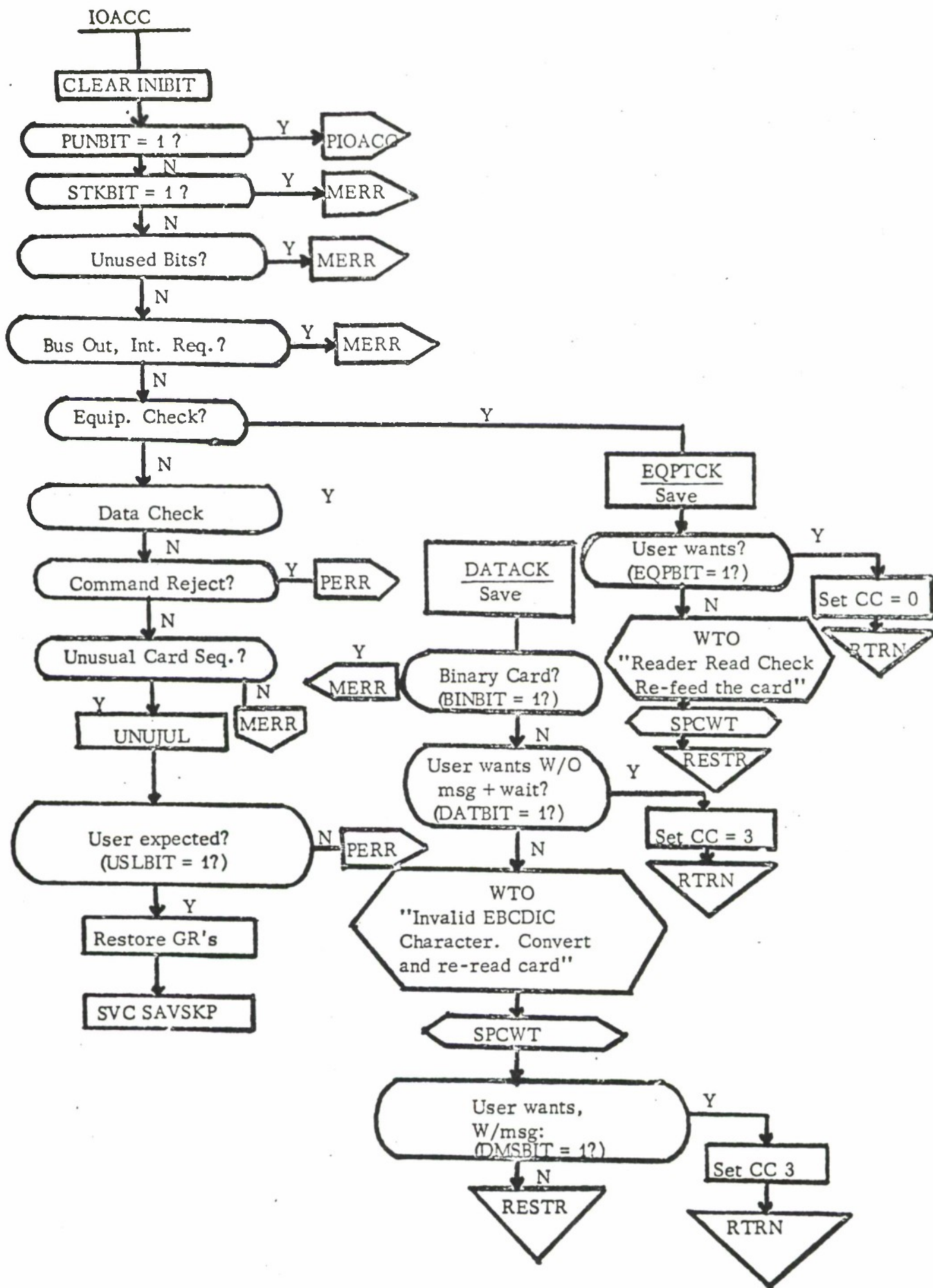
2. The WTO (write to operator) subroutine contains a provision for automatically prefacing a message by the card reader or punch identification if more than one card reader or punch is attached to the system.

3. All combinations of status bits are evaluated; no status bits are ignored. In general, the appearance of a bit in the sense byte which should never be set is treated as a machine error.

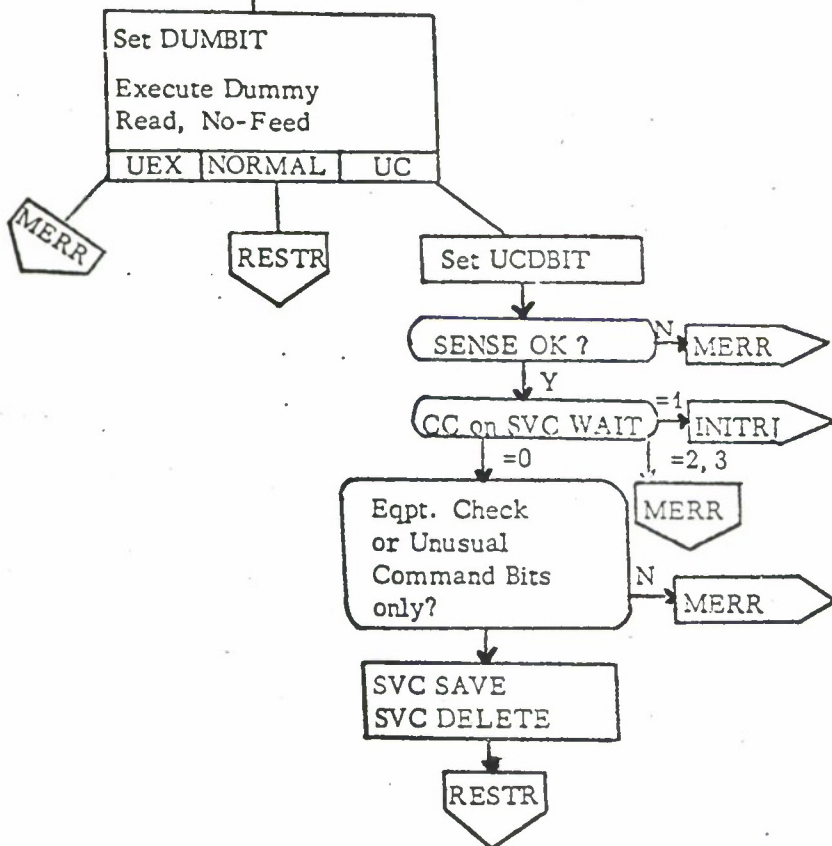
4. Interpretation of the reasons for task termination because of a unit check machine or program error is accomplished by using the console printout of the second word of DSECT2. For each point on the flow chart indicating a branch to MERR, the bits set in HISFLG, MYFLAG, SNSFLG and SENSE form a unique combination. The flow chart can thus be used to determine the cause of the error.

Figure IV.D.2-2. Flow Diagram of CARDUC.





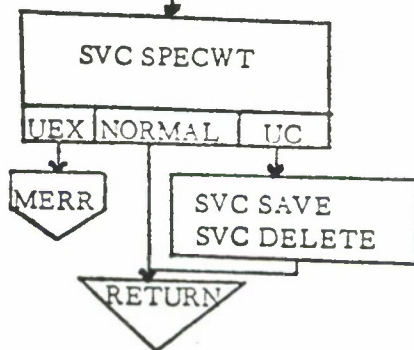
DUMRED



RESTR

Restore GR's
SVC RESTOR

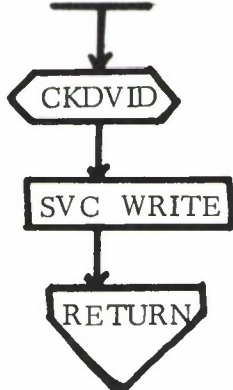
SPCWT



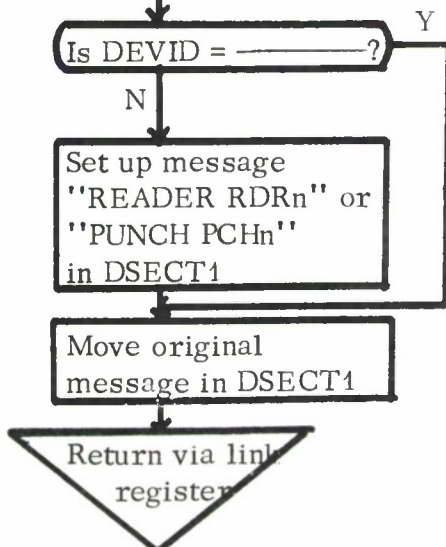
RTRN

Move Branch Inst. to
DSECT1
Restore GR's
Execute Branch

WTO



CKDVID

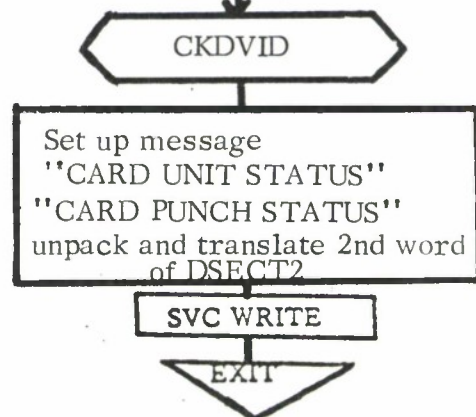


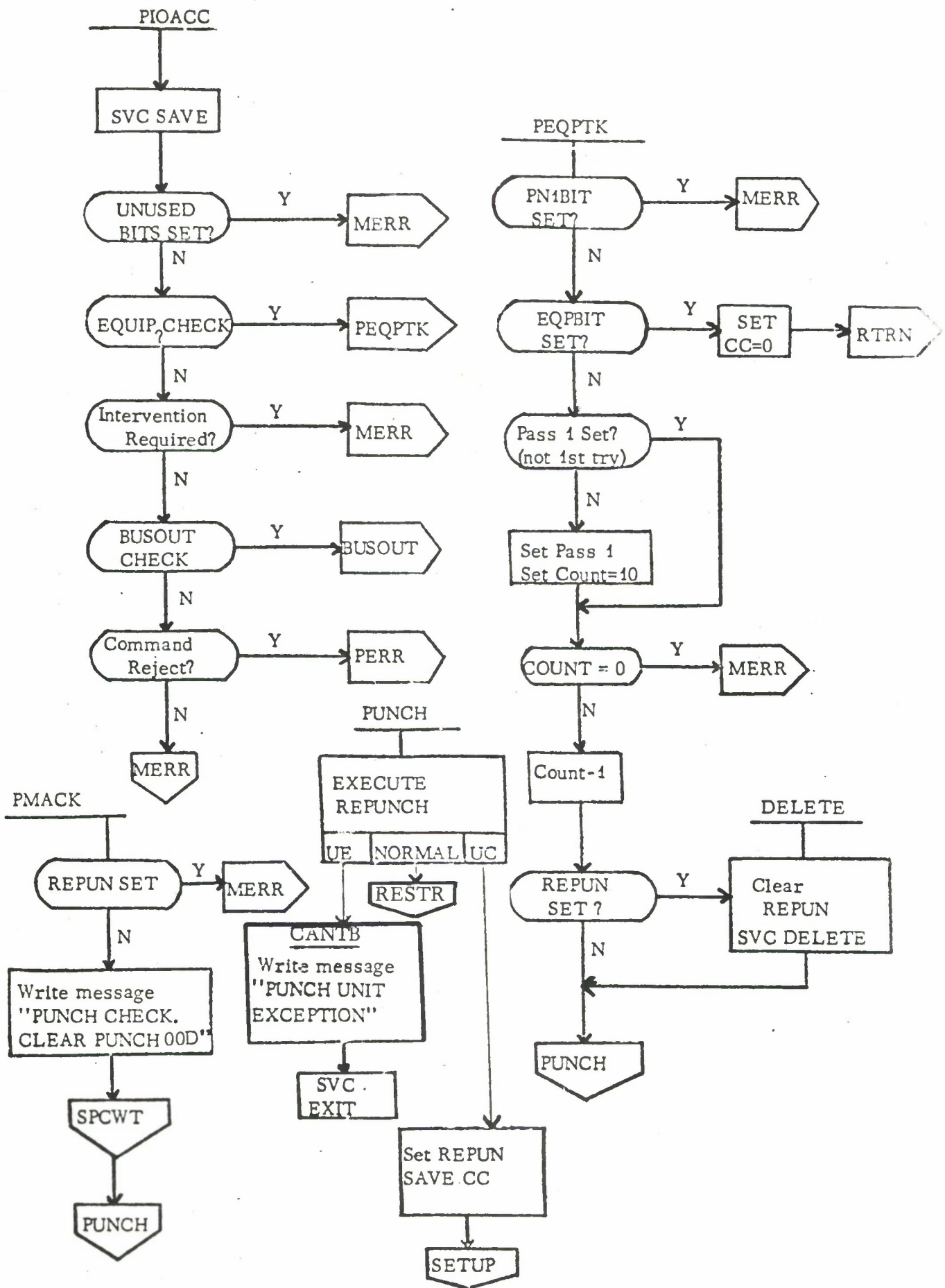
MERR

Set up message
"MACHINE ERROR"

PERR

Set up message
"PROGRAM ERROR"





Lincoln Laboratory Multi-Programming Supervisor

Subroutine Writeup

Tape Unit Check Routines

1 April 1966

General Information:

The tape unit check routine TAPEUC attempts to recover from all causes of tape unit checks. The routine is entered by defining one or both of its entry points as EXTRN's in a job program and using the entry address as the unit check return in the returns list of a tape I/O operation. The entry points are TPRDUC and TPWRUC for tape reading and writing respectively. The routine appears transparent to the job program if the error recovery procedures are successful. If they are not successful in recovering, the routine returns with the condition code set to indicate the error or terminates the job.

Operating Instructions:

Three types of messages can be written by the tape unit check routine. The first type requires operator action and is written when the tape unit is not ready. This situation is transparent to the calling routine and is recoverable when the unit is made ready. The second type of message reflects a condition caused by an operation procedure. The task is terminated but can usually be reinitiated after some appropriate procedure is followed. The third type of message indicates that a permanent error has occurred and either the responsible system programmer or the IBM customer engineer should be notified.

I. Operator Action Message

1. TAPE Txxx NOT READY

When the tape unit is made ready, the task continues.

II. Operator Procedure Messages

2. TAPE Txxx PERM DATA CHECK

This message is written when a permanent read or write error occurs and the recovery procedure is unsuccessful. The task is terminated.

3. TAPE Txxx END OF REEL

This message is written during a write operation when the end-of-tape marker is reached while the tape is being erased to pass over bad tape. The end-of-tape condition is indicated by a unit exception on the erase gap (ERG) operation. The end-of-tape condition during a successful write operation is indicated by a unit exception on the write operation and the unit exception return will be taken. If this message is written, the task will be terminated.

4. TAPE Txxx IS FILE PROTECTED

If an attempt is made to write on a tape unit which is file protected, the above message is written and the task is automatically terminated.

5. NON-EXISTENT TU

This message should not occur, but if it does the responsible system programmer or the IBM customer engineer should be notified. This message might well be printed if the tape unit is physically disconnected from the system.

III. Permanent Error Messages

The information in the following two messages contains a stop code, the condition code from the sense command (cc), and the six sense bytes, in hexadecimal, from the sense command (ssssssssss).

6. PROGRAM ERROR 1 SENSE DATA TAPE Txxx ccssssssssss

This message is printed when the sense bytes after a reading operation are all zero. This may be caused by a calling program command which backspaces to load point. The responsible system programmer should be notified. Other programming errors might occur which are indicated as a machine failure.

7. MACHINE FAILURE e SENSE DATA TAPE Txxx ccssssssssss

When this message is printed, the single character stop code (e) indicates the cause of the error. The codes indicated by an asterisk (*) might occur more frequently than the others and the responsible system programmer should be notified. On the other machine failure conditions the IBM customer engineer should be notified. The stop codes for the machine failure messages are listed below:

Stop Code:

Meaning:

- 0 Sense flag not equal to 1 on TPRDUC. The sense bytes may not be meaningful.
- 2 Overrun on unit after five retries.
- 3 Bus-out check (not on initial selection) after one retry.
- 4 Permanent Bus-out check on initial selection after one retry.
- * 5 Equipment check. May occur when tape is reset during an operation.
- 6 Equipment check. Clock or other serious error.
- * 7 Converter check on TPRDUC operation. A converter check occurs from a 7-track unit if, during a read operation with converter on, the number of characters read cannot be stored in an integral number of bytes. This might be a programming error.
- 8 Intervention required but not on initial selection.
- 9 Sense flag not equal to 1 after initial rejection of a command for a TAPEUC backspace record operation.
- A Permanent bus-out check during a backspace record operation.
- B Sense flag not equal to 1 after initiation of a TAPEUC backspace record operation.
- C Unused sense bytes set during TAPEUC backspace record operation serious I/O error.
- D Zero sense bytes but tape not at load point.
- E Sense flag not equal to 1 after initial rejection of a command for a TAPEUC forward space record operation.
- F Permanent bus-out check during a forward space record operation.
- G Sense flag not equal to 1 after initiation of a command for a TAPEUC forward space record operation.

Stop Code:

Meaning:

H	Sense Status A = Status B. Serious machine error.
* I	Non-existent tape unit. This message follows message 5.
J	Sense flag not equal to 1 on TPWRUC. The sense bytes may not be meaningful.
K	Sense bytes on TPWRUC indicate a condition other than intervention required or Bus-out check on initial selection. Serious error.
L	Unused sense bits set after a TPWRUC operation.
M	Zero sense bytes after a TPWRUC operation.
* N	Command reject from a TPWRUC operation but tape is not file protected. This might be a programming error.
O	Sense flag not equal to 1 after an initial rejection of a ERG command.
P	Permanent bus-out check during an ERG operation.
Q	Sense flag not equal to 1 after initiation of an ERG command.
R	Unit check with other than an equipment check or data check condition on an ERG command during a TPWRUC operation.

Programming Instructions:

This routine handles tape unit checks for two types of commands:

1. Tape reading (entry point TPRDUC)
2. Tape writing (entry point TPWRUC).

Linkage from the calling program requires that the entry point address, which has been defined as an EXTRN in the calling program, be defined as the unit check return in the returns list of a SVC STIO. The calling program must also initialize general register 13 to point to a double-word aligned storage area, DSECT1, depicted in Figure IV.D.3-5. The first word in DSECT1 must point to a second double-word aligned storage area, DSECT2, as illustrated below. DSECT1 is used as a work area by the unit check routine and must be 80 bytes long for tape reading and 88 bytes long for tape writing. DSECT2 contains parameters used by the routine and is 5 words (20 bytes) long.

GR 13 contains the address of DSECT1

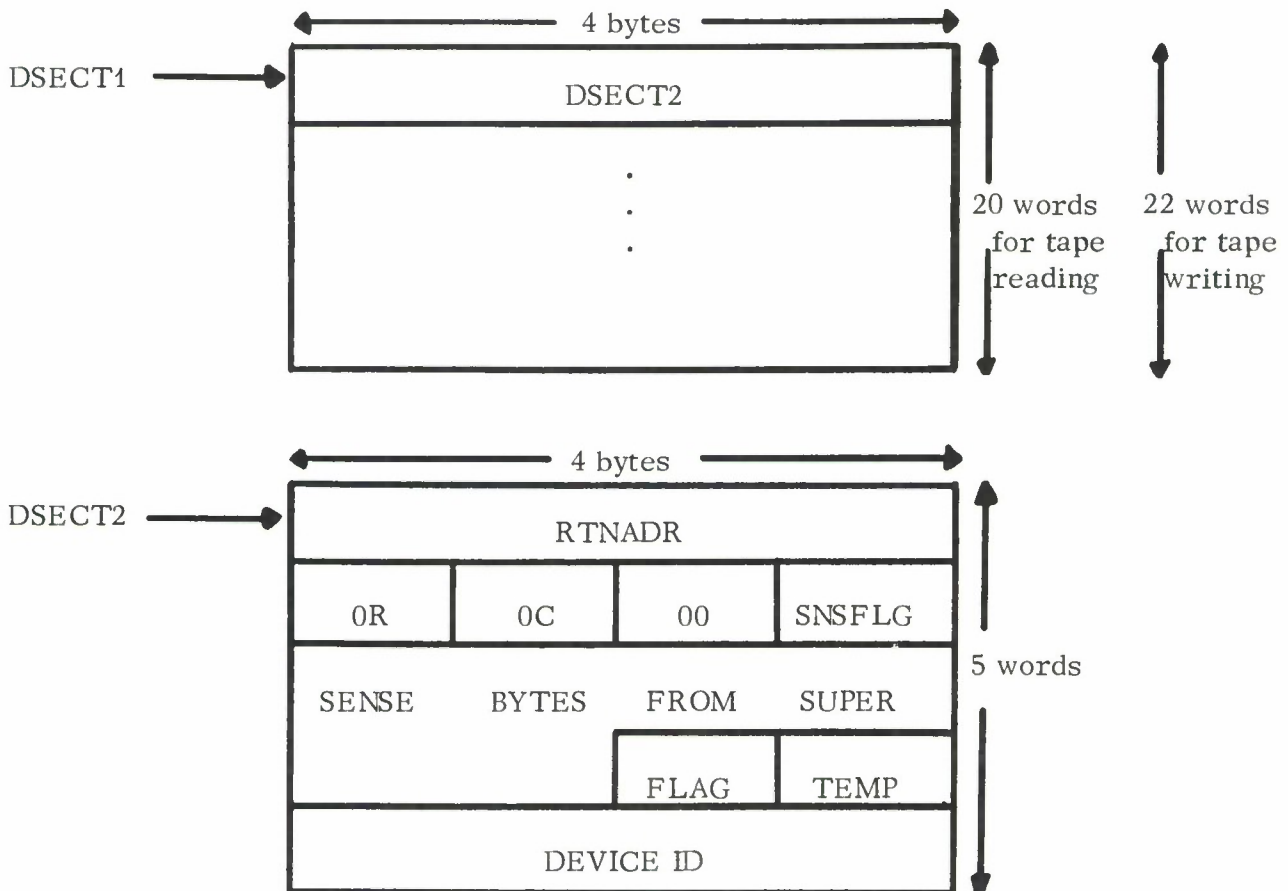


Figure IV.D.3-1. DSECTs used in tape unit check routine.

If any of the unit check return options are selected by the FLAG bits, the calling program must place the return address into the first word of DSECT2. RTNADR is the location in the calling program to which control will be returned with the condition set to indicate the cause of the return. This full word in DSECT2 must be reserved even if the calling program does not use any of the options.

The "retry" count, R, specifies the number of times the operation is retried on an error condition and the "cleans" count, C, specifies the number of times the tape cleaning procedure is performed. After each cleaning procedure, the operation may be retried the number of times specified by the retry count. The maximum number of retries or cleans is 15.

The SNSFLG and SENSE byte areas are used by the supervisor to provide sense information to the unit check routine. Prior to issuing the first SVC STIO, the location of the SNSFLG byte must be passed to the supervisor via an SVC SNSADR to initiate automatic sensing for the tape operation. SVC GETID must also be issued and the device ID obtained placed in the fifth word of DSECT2.

The FLAG byte is used to select unit check routine options. Bits 5, 6, and 7 indicate whether the TAPEUC routine should return to the calling program on an abnormal unit check condition or if the job should be terminated. If an abnormal unit check condition occurs and the appropriate FLAG bit is set, the routine returns to the calling program with the condition code set to indicate the cause of the error.

The meaning of the FLAG bits (numbered 0-7 from left to right) and the condition code setting on return from the unit check routine are summarized and described below. If a bit is not set and the indicated abnormal unit check condition occurs, the indicated message will be printed and the task terminated.

On Reading

bits:	0	1	2	3	4	5	6	7	Condition Code:	Message:
4	-									position tape at beginning of error record on permanent read error
5	-								1	2
6	-								2	7
7	-								3	7
										equipment check and not ready together

On Writing

bits:	0	1	2	3	4	5	6	7	Condition Code:	Message:
5	-								1	2
6	-								2	3
7	-								3	7
										equipment check and not ready together

When a permanent read error occurs, the tape can be positioned either before or after the record in error. This is determined by the setting of bit 4. If bit 5 is set and the recovery procedure for a redundancy error is unsuccessful, a return will be made to the calling program with condition code 1; if bit 5 is not set, message 1 will be printed and the task terminated.

If bit 4 is set, the tape will be positioned at the beginning of the record which caused an unrecoverable error condition before return is made to the calling routine. If the bit is off, the tape will be positioned after the bad record. If a job is to change modes after reading a record in error, then bit 4 should be set.

If bit 5 or 6 is not set and the indicated abnormal unit check condition occurs, the appropriate message is printed and the task is terminated. If bit 5 or 6 is set and the indicated abnormal unit check condition occurs, a return is made to the calling program with the appropriate condition code set.

An equipment check and not ready condition occurring together indicates that the tape became not ready while a reading or writing operation was in progress and the position of the tape is uncertain. If bit 7 is set and the tape unit becomes not ready during an operation, the not ready message will be printed and when the tape is made ready, a return will be made to the calling routine with the condition code set to 3. If bit 7 is not set and the tape unit becomes not ready during an operation, the equipment check message will be printed out and the task will be terminated.

The byte shown as TEMP is used and altered by the unit check routine but may also be used by the calling routine.

Before each SVC WAIT for a tape operation, general register 13 must point to DSECT1 and the retry count, clean count, and zero byte (byte after clean count) must be reset. The TAPEUC routine does not alter any GR's including GR 13, except when a return to RTNADR on an abnormal unit check condition is made, GR 13 will be altered.

Error Recovery Procedures:

The tape unit check routine tries to recover from an error condition (data check) whenever possible. The retry procedure for a reading error consists of backspacing over the error record and rereading the record. The retry procedure for a write error consists of backspacing over the error record, skipping tape for a length of 3-1/2 " (using the erase record gap command) and then rewriting the record. The number of times the above retry procedure is performed is specified by the "retry" count parameter of the unit check routine. If successive retries do not eliminate the error, the tape can be made to pass through the cleaning heads of the tape unit. This is performed after the last retry by backspacing the tape 5 records and then forward spacing the tape 4 records. This procedure repositions the tape after the last good record or at load point. The number of times the cleaning procedure is performed is specified by the "cleans" count parameter of the unit check routine. After each cleaning procedure, the operation

may be retried the number of times specified by the retry count. The maximum number of retries or cleans is 15.

It should be noted that before each retry on a write operation the tape is spaced forward 3-1/2". Thus, after 10 retries 35 inches of blank tape will have been spaced over.

When recovery from a permanent error fails, the tape is positioned beyond the last good record after a write operation and is positioned either before or after the record in error after a read operation. The tape is positioned after the record in error on a read operation if flag bit 4 was not set and is positioned before the record in error if flag bit 4 was set. If a job is to change modes after reading a record in error, then flag bit 4 should be set.

Sample Program:

An example of the use of the tape unit check routine is illustrated by the sample coding below, which uses the tape read unit check entry. The number of REREAD's is equated to 8 and the number of CLEANS is equated to 2.

EXAMPL	START	0	
	EXTRN	TPRDUC	or TPWRUC
	...		
	LA	0, LDN	
	LA	1, SNSFLG	
	SVC	SNSADR	Set up automatic sense
	SVC	GETID	Get the device ID
	ST	0, DEVID	Store device ID for TAPEUC
	...		
	LA	13, DSECT2	First word of DSECT1 work area contains address of TPRDUC parameters
	ST	13, DSECT1	Set return address in parameter area
	LA	13, RTNADR	
	ST	13, DSECT2	
	MVI	FLAG, BITS	Set flag bits
	...		
	LA	0, LDN	
	LA	2, RTNLST	
	LA	13, DSECT1	Set GR 13
	MVC	COUNT(3), COUNTS	Initialize count before all reads or writes
	SVC	STIO	
	SVC	WAIT	
	...		

RTNLST	DS	F	
	DC	A(UE)	Unit exception return
	DC	A(TPRDUC)	Unit check return
	...		
	DS	0D	
DSECT1	DC	A(DSECT2)	Work area for TAPEUC
	DS	19F	Work area for TPRDUC
	...		(DS 21F for TPWRUC)
	DS	0D	
DSECT2	DC	A(RTNADR	Parameters for TPRDUC or TPWRUC
COUNT	DS	3C	
SNSFLG	DS	C	
SENSE	DS	6C	
FLAG	DS	C	
TEMP	DS	C	
DEVID	DS	F	
	...		
COUNTS	DC	AL1(REREAD)	No. times record should be reread
	DC	AL1(CLEANS)	No. times tape should be cleaned
	DC	AL1(0)	This byte must be zero
REREADS	EQU	8	Must be less than 16
CLEANS	EQU	2	Must be less than 16

Lincoln Laboratory Multi-Programming Supervisor

Subroutine Write-Up

Printer Unit Exception and Unit Check Routine

General Information:

The printer unit exception and unit check routine is entered by defining one or both of its entry points as EXTERN'S in a calling program and using the entry address as a unit exception return and/or a unit check return in the returns list of a SVC STIO for the printer. A unit check indicates that the printer or its controls require program or operator intervention. A unit exception indicates that a hole has been sensed in Channel 12 of the carriage control tape. The entry points are PRTUE (unit exception) and PRTUC (unit check).

The printer unit exception and unit check routine is explained in three parts:

1. Operating instructions describing console messages produced by the routine and required operator actions.
2. Programming instructions describing how to use the routine in writing programs.
3. Program description including a flow chart of the routine.

Operating Instructions:

The printer unit exception and unit check routine writes two types of messages. The first type requires operator intervention indicating that the printer is not ready resulting from one of the following: (1) a sync check, (2) an end-of-forms condition, (3) a forms check, (4) an opened mechanical interlock, or (5) the stop key being pressed. The second type indicates that a permanent error has occurred and the task terminates. The operator should then notify either the IBM customer engineer or the responsible system programmer.

I: Operator Action Messages

These messages are in two forms, depending on the number of printers attached to the system. If there is more than one printer, the message is preceded by the printer identification, "PRINTER PTRn". If there is only one printer, the message appears by itself.

1. PRINTER NOT READY

This message indicates that the printer ready light is not on due to the occurrence of one of the above mentioned conditions. The task continues when the printer is made ready.

II: Permanent Error Messages

There are two types of permanent error messages, one involving a program error in which case the system programmer should be notified, and the other involving machine malfunction in which case the customer engineer should be notified. The task terminates on any permanent error condition. The status information is displayed and consists of three bytes of hexadecimal information from the routine (see Programming Instructions). The meaning of the bytes are as follows:

sc	stop code
sf	sense flag SNSCC
sb	printer sense byte SENBYT

The status information is preceded by the printer identification.

---	if only one printer attached to system
PTRn	if more than one printer is attached.

2. PRINTER EQUIPMENT CHECK

MACHINE FAILURE PRINTER SENSE DATA ---- scsfbs

A printer equipment check has occurred after one retry attempt. Despite the equipment check, the line of information is printed. The routine writes the message, "ABOVE LINE PRINTED UNDER MACHINE MALFUNCTION", on the printer.

3. BUS-OUT CHECK

MACHINE FAILURE PRINTER SENSE DATA ---- scsfbs

A bus-out check has occurred after one retry attempt. Despite the bus-out check, printing is not suppressed. The routine writes the message, "ABOVE LINE PRINTED UNDER MACHINE MALFUNCTION", on the printer.

4. PROGRAM ERROR PRINTER SENSE DATA ---- scsfbs

The responsible system programmer should be notified when this message is printed on the console typewriter.

The stop codes for the machine failure and program error messages are listed below:

<u>Stop Code</u>	<u>Meaning</u>
0	Sense flag not equal to 1 on PRTUC
1	Unused bits set in sense byte. Machine error.
2	Equipment check.
3	Intervention required
4	Bus-out check (not on initial selection) after one retry.
5	Bus-out on initial selection after one retry.
6	Channel 9 check.
7	Command reject on initial selection. Program error.
8	Command reject (not on initial selection). Program error.
9	Unit check on attempting to write a message on the printer.
A	Unit exception on SVC SPWAIT.
B	Unit check on eject page command.

Programming Instructions:

The routine handles both printer unit exceptions and printer unit checks. Linkage from the calling program requires placing the proper entry point address PRTUE, which has been defined as an EXTRN in the calling program, into the returns list as the unit exception address; and the entry point address PRTUC, which also has been defined as an EXTRN in the calling program, into the returns list as the unit check address. The calling program must also initialize general register 13 to point to a double word aligned storage area DSECT1, depicted in Fig. IV.D.4-1. The first word in DSECT1 must point to a second double-word aligned storage area DSECT2, also shown in Fig. IV.D.4-1. DSECT1 is used as a work area by the routine and must be 80 bytes long. DSECT2 contains the parameters used by the routine and is 16 bytes long. GR 13 contains the address DSECT1.

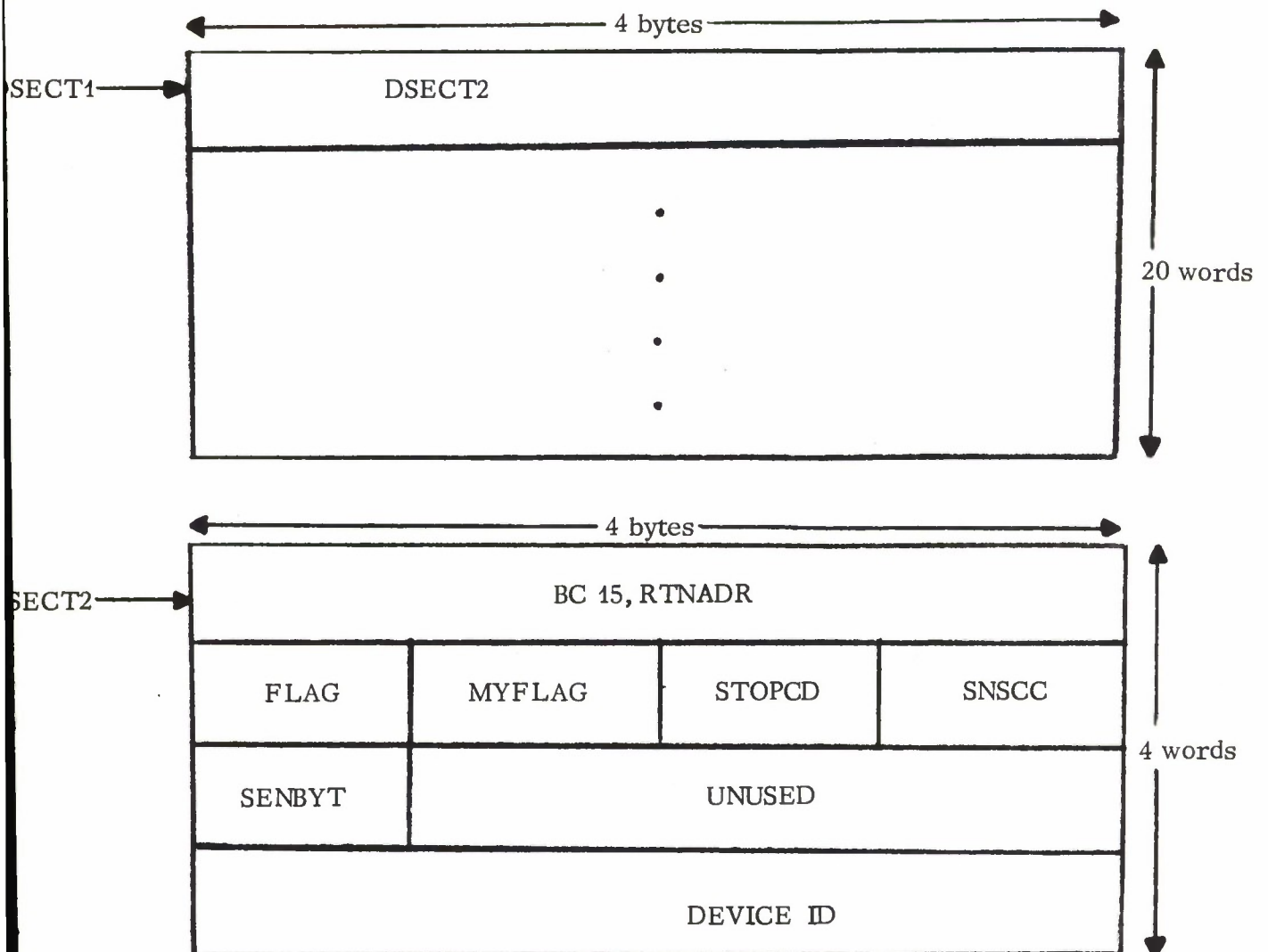


Fig. IV.D.4-1. DSECTs used in printer unit exception and unit check routine.

If any unit exception or unit check return option is selected, the calling program must place the assembled instruction "BC 15, RTNADR" into the first word of DSECT2. RTNADR is the location in the calling program to which control can be returned with the condition code set to 1 on a Channel 9 check or set to 3 on a Channel 12 check. This full word in DSECT2 must be reserved even if the calling program does not use the option.

The second word contains four bytes of information. The FLAG byte contains information passed from the calling program. The FLAG byte is used to select the printer unit exception and unit check options and is not altered by the routine. The meaning of the FLAG bits (numbered 0-7 from left to right) are as follows:

Bits: 01234567

- 5 - If this bit is set, a page is ejected when a hole is sensed in Channel 9 of the carriage control tape during a carriage space operation. If this bit is not set, no ejection occurs.
- 6 - If this bit is set, a page is ejected when a hole is sensed in Channel 12 of the carriage control tape during a carriage space operation. If this bit is not set, no ejection occurs.
- 7 - When this bit is set, a return is made to the job program on a check condition. The condition code is set to 1 on a Channel 9 check or set to 3 on a Channel 12 check. If this bit is not set, the program continues normally.

These bits must all be set or cleared by the program as desired. The MYFLAG and STOPCD bytes are used by the routine and must be cleared prior to each SVC WAIT issued for the printer in the calling program.

The SNSCC and SENBYT bytes are used by the supervisor to provide sense information to the routine. The location of the SNSCC byte must be passed to the supervisor via an SVC SNSADR to initiate automatic sensing prior to issuing the first SVC STIO for the printer. SVC GETID must also be issued and the device ID obtained and placed in the fourth word of DSECT2.

Sample Program:

An example of the use of the printer unit exception and unit check routine is shown in Fig. IV.D. 4-2. The sample coding uses both the unit exception and unit check entries.

Program Description:

The flow chart of the printer unit exception and unit check routine reveals the structure of the routine. There are separate entries for unit exceptions and unit checks. The flow chart is given in Fig. IV.D. 4-3.

SAMPLE	START	0	
	EXTRN	PRTUE	
	EXTRN	PRTUC	
	...		
	LA	13, DSECT1	
	LA	0, LDN	Put logical device number of the printer in
	LA	1, SNSCC	GR0 and set up automatic sensing
	SVC	SNSADR	
	SVC	GETID	Get device ID
	ST	0, DEVID	
	MVI	FLAG, X'06'	Set FLAG bit to eject the page on a Channel 9 or
	...		a Channel 12 check and continue with normal
	...		sequencing.
	...		
PRINT	LM	0, 2, PRTLST	
	SVC	STIO	
	MVI	MYFLAG, X'00'	Clear MYFLAG
	MVI	STOPCD, X'00'	and STOPCD
	SVC	WAIT	
	...		
PRTLST	DC	F'1'	
	DC	A(PRTCCW)	
	DC	A(RTNLST)	
	...		
RTNLST	DS	F	
	DC	A(PRTUE)	
	DC	A(PRTUC)	
	...		
DSECT1	DS	0D	
	DC	A(DSECT2)	
	DS	19F	
	...		
DSECT2	DS	0D	
FLAG	BC	15, RTNADR	
MYFLAG	DS	C	
STOPCD	DS	C	
SNSCC	DS	C	
SENBYT	DS	F	
DEVID	DS	F	

Fig. IV. D. 4-2. Sample code showing the use of the PRNTUC routines.

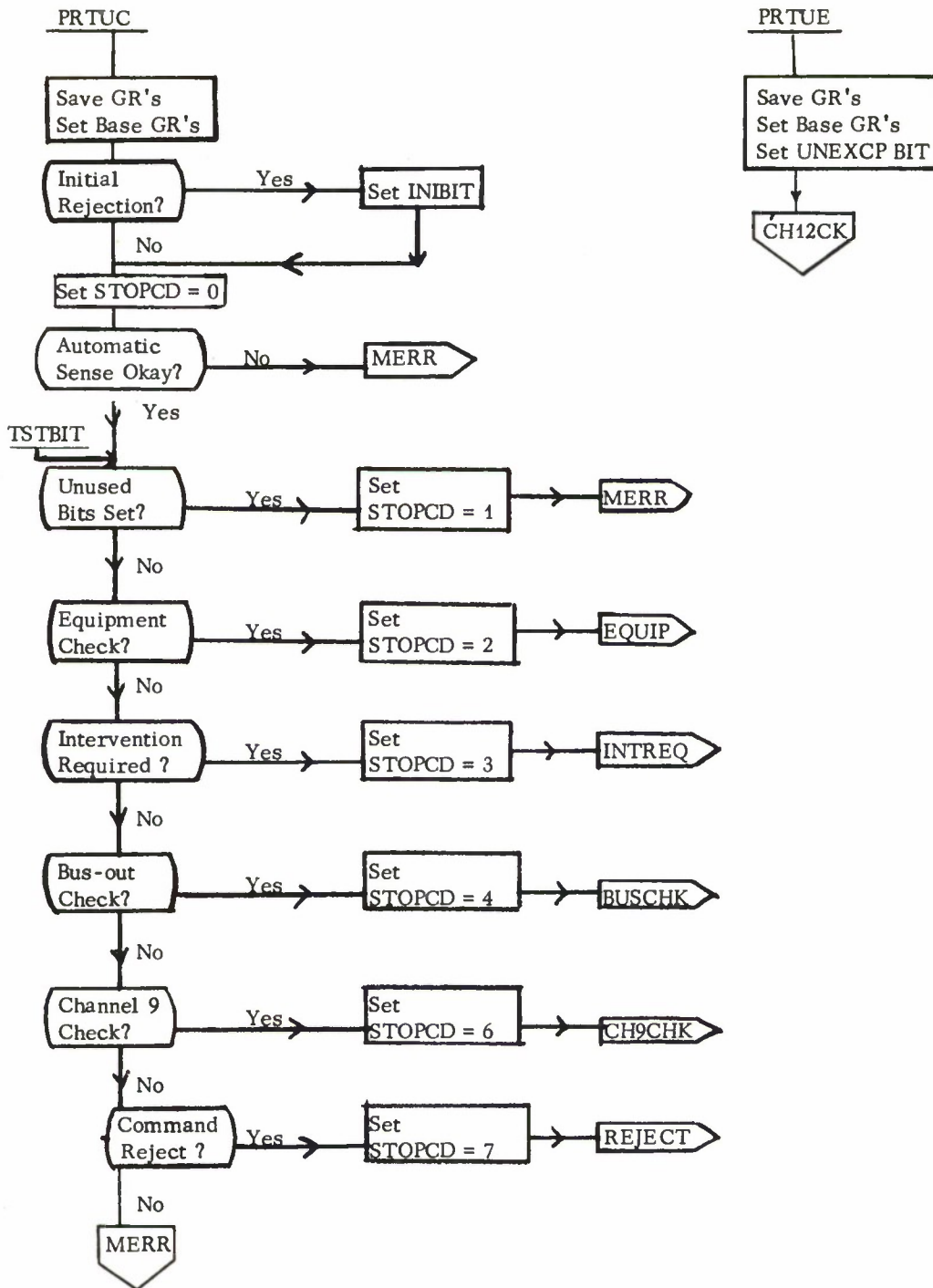


Fig. IV. D. 4-3. Flow diagram of PRNTUC

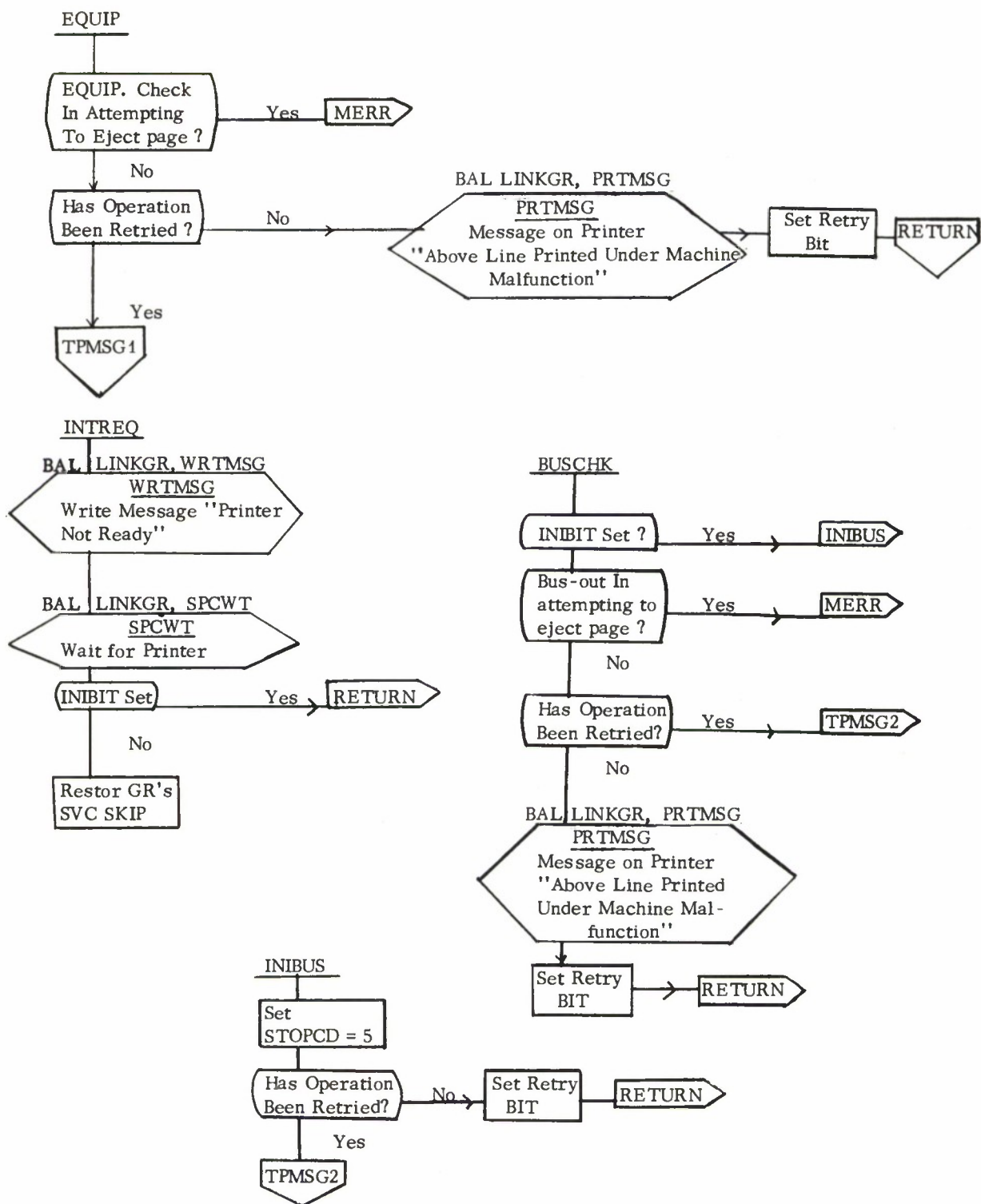


Fig. IV. D. 4-3. Flow diagram of PRNTUC. (Cont.)

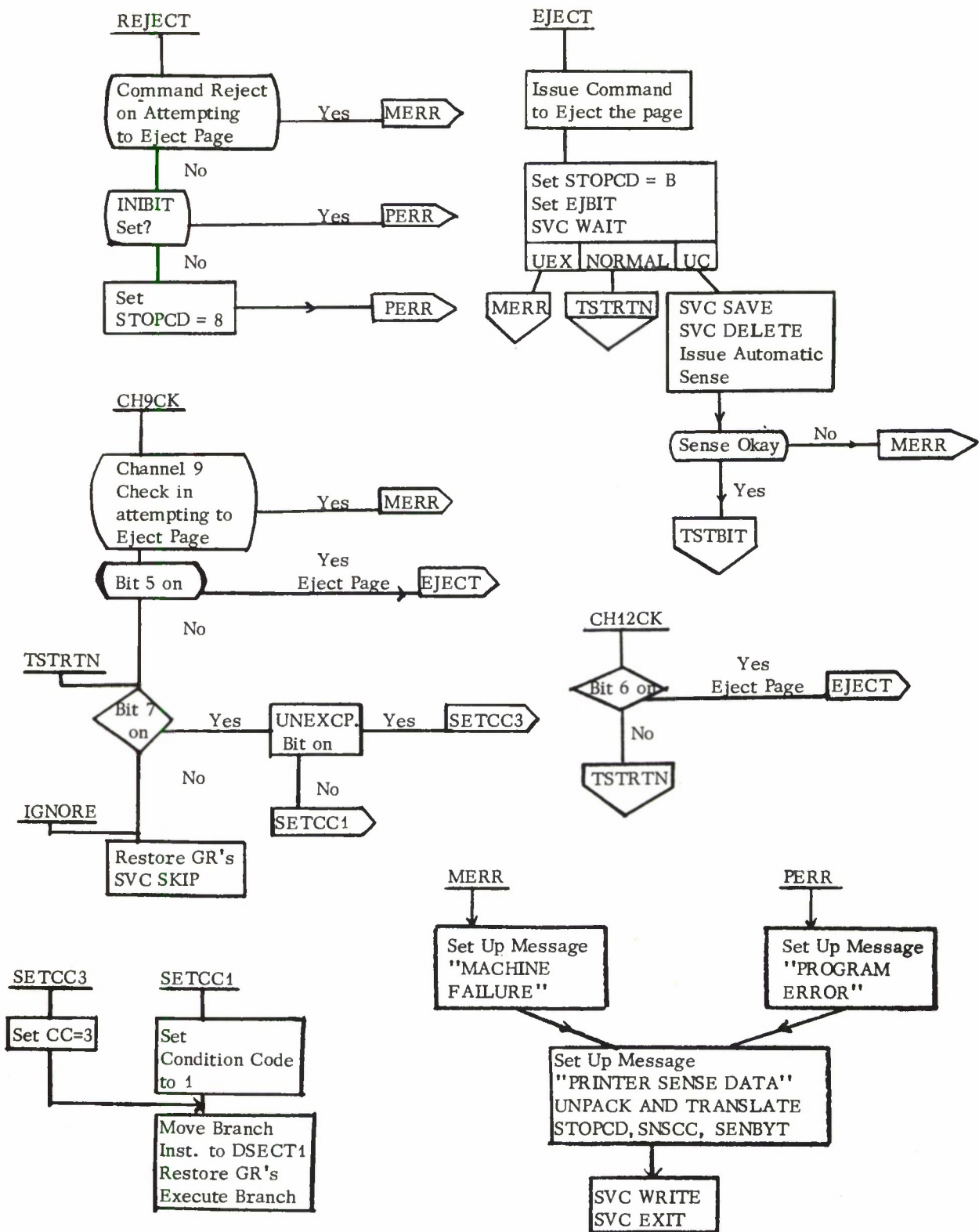


Fig. IV. D. 4-3. Flow diagram of PRNTUC. (Cont.)

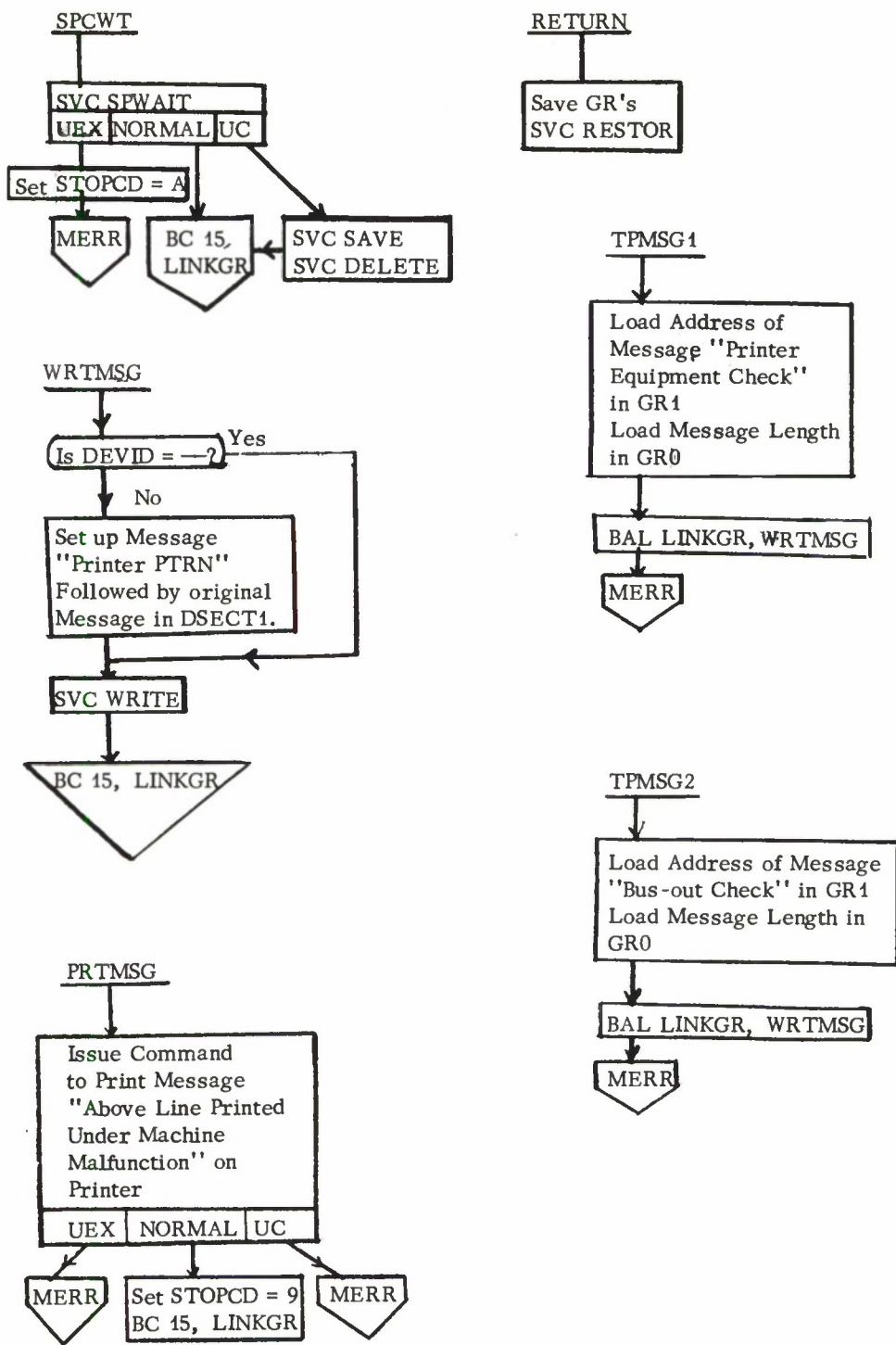


Fig. IV. D. 4-3. Flow diagram of PRNTUC. (Cont.)

Lincoln Laboratory Multi-Programming Supervisor

Subroutine Write-Up

Find Job List Subroutine

1 June 1966

General Information:

The re-entrant system subroutine, FJBLST, looks for a job list entry (name of and entry to an LLMPS job program) in the supervisor table JOBLST.

Upon entering the routine, FJBLST expects a job number in GR0 and the address of a job name in GR1. If FJBLST is successful in locating the job list entry, it returns the location of the job list entry in GR2 and sets the condition code to 1, indicating a non re-entrant job, or to 3, indicating a re-entrant job. If FJBLST is unsuccessful, it returns with the condition code set to 0.

Calling Sequence:

The user must define the FJBLST routine as an external symbol in his program and call it by a BALR 14, 15 instruction where GR15 is loaded to contain the address of the FJBLST routine.

General register 14 and 15 must be preset prior to entering FJBLST. GR14 is used as the return register to the calling program and GR15 is used as the base register.

SAMPLE Program:

An example of the use of FJBLST is shown below:

CHGBUF	START	0	
	EXTRN	FJBLST	
	.		
	.		
	.		
SUBRTN	L	0, JOBNUM	
	LA	1, JOBNME	
	L	15, ADFJBL	
	BALR	14, 15	
TSTCC	BC	8, NOTFND	Job not found in job list
	BC	4, OKAY	Job found in job list
	BC	1, OKAY	Re-entrant job found in job list
OKAY	L	2, 12(2)	Address of job (4th word in job list)
	.		
	.		
	.		

NOTFND

SVC

EXIT

ADFJBL

DC

A(FJBLST)

JOBNUM

DC

F'0'

task 0 indicates job not active.

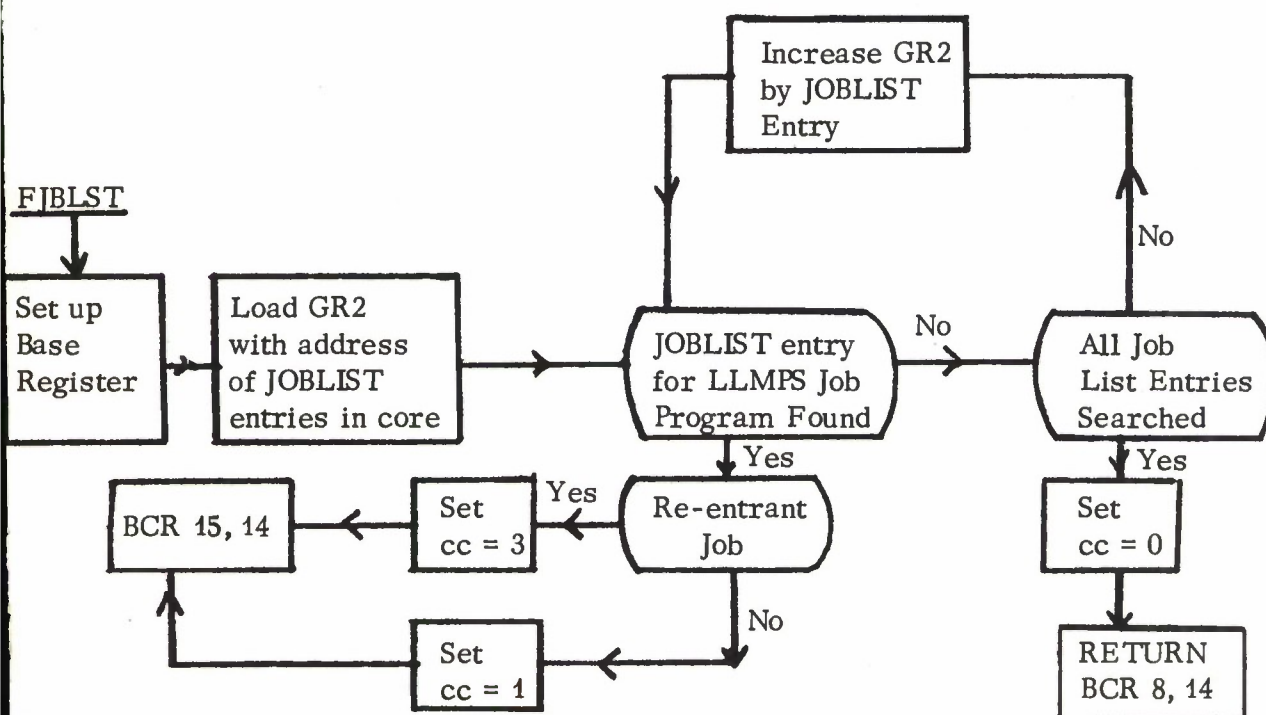
JOBNAME

DC

C'T7DUMP'

Program Description:

The flow chart of the routine, FJBLST, is shown below



UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Lincoln Laboratory, M.I.T.

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

None

3. REPORT TITLE

Lincoln Laboratory Multi-Programming Supervisor

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Lincoln Manual

5. AUTHOR(S) (Last name, first name, initial)

Belvin, Frank Winett, Joel M.

6. REPORT DATE

6 January 1967

7a. TOTAL NO. OF PAGES

232

7b. NO. OF REFS

None

8a. CONTRACT OR GRANT NO.

AF 19(628)-5167

b. PROJECT NO.

649L

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

Lincoln Manual 78

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

ESD-TR-67-14

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

None

12. SPONSORING MILITARY ACTIVITY

Air Force Systems Command, USAF

13. ABSTRACT

A supervisor program has been written for the IBM System/360 which allows the multi-programming of several jobs, each with its own input-output requirements.

14. KEY WORDS

supervisor program
IBM System/360
multi-programming

UNCLASSIFIED

Security Classification